Solving ill-posed Image Processing problems using Data Assimilation

Dominique Béréziat, Isabelle Herlin

▶ To cite this version:

HAL Id: inria-00538510

https://hal.inria.fr/inria-00538510

Submitted on 22 Nov 2010

Solving ill-posed Image Processing problems using Data Assimilation

Dominique Béréziat · Isabelle Herlin

Abstract Data Assimilation is a mathematical framework used in environmental sciences to improve forecasts performed by meteorological, oceanographic or air quality simulation models. It aims to solve an evolution equation, describing the temporal dynamics, and an observation equation, linking the state vector and observations. In this article we use this framework to study a class of ill-posed Image Processing problems, usually solved by spatial and temporal regularization techniques. An approach is proposed to convert an ill-posed Image Processing problem in terms of a Data Assimilation system, solved by a 4D-Var method. This is illustrated by the estimation of optical flow from a noisy image sequence, with the dynamic model ensuring the temporal regularity of the result. The innovation of the paper concerns first, the extensive description of the tasks to be achieved for going from an image processing problem to a data assimilation description; second, the theoretical analysis of the covariance matrices involved in the algorithm; and third a specific discretisation scheme ensuring the stability of computation for the application on optical flow estimation.

Keywords computer vision, inverse problems, data assimilation, non linear advection, optical flow.

1 Introduction

In the research field of Image Processing, most problems are ill-posed (according to Hadamard definition) in the sense that it is not possible to provide a unique solution [4]. 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

D. Béréziat
Université Pierre et Marie Curie, LIP6, 4 place Jussieu, 75005 Paris, France.
E-mail: Dominique.Bereziat@upmc.fr
Tel.: +33 1 44 27 47 71 Fax.: +33 1 44 27 53 53

I. Herlin
INRIA, CEREA, Joint Laboratory ENPC-EDF R&D, Université Paris-Est,
Domaine de Voluceau, Rocquencourt - B.P. 105, 78153 Le Chesnay, France.

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 with a different result.

An usual strategy to solve ill-posed problems is to provide additional information to the Image Model. Two options may be considered. 1) Explicit information: complementary images are used to enlarge the set of input data. However, this is generally not possible because additional acquisitions, having the requested properties, are not available. 2) Implicit information such as hypotheses on image properties or constraints on the solution. A usual way 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, which is called "Tikhonov regularization method" in the literature [22]. In the general case, these additional properties or constraints are expressed as equations, which, combined to the Image Model, lead to a new invertible model.

Assuming not only one or two images but a whole sequence is available, enhancement may be obtained by taking into account the temporal evolution. Let us illustrate this on the image segmentation issue, which is a pure static problem. A spatial regularization method, such as Shah-Mumford's functional [10], produces a segmentation which is a compromise between a spatially smooth solution and the adequacy to the input data. If the segmentation process is performed directly on the whole sequence, following [26], the solution is then seek as a function depending on the spatial and temporal coordinates. This space-time 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 space-time domain, this leads to an high computing complexity compared to a pure spatial model, which is a limiting factor for operational applications.

As the space-time approach fails in case of missing data, an alternative is to consider the temporal dynamics of the data. One challenge becomes to write an efficient dynamic model and to include it in the solution computation. Such information can be inferred, for instance, from *a priori* knowledge on the observed phenomena and related images. Moreover, data quality has to be evaluated in order to ignore missing data in the process. The first innovation of the paper is thus to use the dynamic model and the data quality measure to rewrite the image processing problem in a *Data Assimilation* system [8] in a generic way.

A Data Assimilation method solves a system of three equations with respect to a state vector, corresponding to the studied quantities:

— an evolution equation describes the evolution of the state vector over time, using an operator called the "evolution model";
— an "observation equation" models the links between the state vector and the observations provided by the image sequence;

– the initial condition of the state vector.

Each equation of the system only approximates the reality and a description of the error is stated in terms of a Gaussian noise characterized by a covariance matrix. For a given error, such matrix depicts the dependencies between error's coordinates on the one hand, and between two different locations in the space-time domain on the other hand.

We aim to use Data Assimilation as a generic framework to solve ill-posed Image Processing problems. The spatio-temporal regularization constraint is replaced by an evolution equation. Image dynamics being correctly approximated by this equation, the first drawback of the space-time approach is no more occurring. Describing the tasks to be achieved for going from an ill-posed image processing problem to a data assimilation system is the first contribution of this paper. A study of covariance matrices and their impact during the energy minimization has been performed. We show how a relevant choice allows to deal with missing data and solve the second drawback of the space-time approach. This is the second contribution of this work. These general considerations are illustrated on the problem of optical flow estimation. The third contribution is to propose a robust and innovative numerical scheme to implement the transport of velocity by itself, used in this paper as evolution equation. We show that the computation of the state vector can then be done locally in time, with specific choices for covariance matrices. This allows the algorithm to work on a frame-by-frame basis, unlike the space-time approach.

This article is organized as follows. 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 can Data Assimilation be used to solve ill-posed problems by assimilating images in an appropriate evolution model is explained in Section 4. Section 5 is then a direct application describing how to compute optical flow in this framework, using a robust numerical scheme. It 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 gave the following definition: a problem is well-posed if 1) it has a unique solution, 2) the solution depends continuously on the input [4]. 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 well known ill-posed image processing problems because the equations used for modeling them are under-constrained (see the aperture problem of optical flow for instance).

Links between image properties and the solution are modeled as a set of equations constituting the so-called 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 mathematically written as:

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

with $\mathbb{I}$ a differentiable 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 for 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 considered 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 contour or $\mathbb{I}(\mathbf{X}, Y_1, Y_2)(\mathbf{x}) = Y_2(\mathbf{x} + \mathbf{X}(\mathbf{x})) - Y_1(\mathbf{x})$ for estimating the optical flow $\mathbf{X}$ between two images 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 $\Psi(0) = 0$. As Equation (1) is under-constrained, a possible method to obtain a unique solution is to use the 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 $\alpha_n = 0$ for $n \neq 1$ ensuring a first order regularization or $\alpha_n = 0$ for $n \neq 1, 2$ ensuring a second order regularization. The norm used in the regularizing term is often the Euclidean norm but other choices are possible: for instance, to determine optical flow, Nagel uses in [11] an oriented norm driven by the local image configuration. The regularization performed by the algorithm is then weak 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 $\mathbf{X}$. With a first order regularization, $\frac{\partial}{\partial \mathbf{x}}$ is usually denoted ∇, and $E(\mathbf{X}) = \int (\Psi(\|\mathbb{I}(\mathbf{X}, \mathbf{Y})\|) + \alpha_1 \Psi(\|\nabla \mathbf{X}\|)) d\mathbf{x}$. The general expression of the corresponding Euler-Lagrange equation becomes:

$$\Psi'(\|\mathbb{I}(\mathbf{X}, \mathbf{Y})\|) \frac{\partial \mathbb{I}}{\partial \mathbf{X}} - \alpha_1 \nabla . \left(\Psi'(\|\nabla \mathbf{X}\|) \frac{\nabla \mathbf{X}}{\|\nabla \mathbf{X}\|} \right) = 0 \tag{3}$$

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 $\mathbf{Y}$ is considered as only depending on the spatial coordinate $\mathbf{x}$ and the Image Model $\mathbb{I}$ is then static. In the case of optical flow computation, at least two frames are required to compute the temporal derivatives, but the result obtained on one image, using (3), has no link with those obtained on adjacent frames. An improved solution, when dealing with a temporal sequence, has been proposed by Weickert *et al* [26] and consists in minimizing the functional:

$$E(\mathbf{X}) = \int_{\Omega} \int_0^{\mathbf{T}} \left(\Psi(\|\mathbb{I}(\mathbf{X}, \mathbf{Y})\|) + \sum_{n \geq 0} \alpha_n \Psi \left(\left\| \frac{\partial^n \mathbf{X}}{\partial \mathbf{x}^n} \right\|, \left\| \frac{\partial \mathbf{X}}{\partial t} \right\| \right) \right) d\mathbf{x}dt \tag{4}$$

with $\mathbf{X}$ and $\mathbf{Y}$ becoming space-time dependent. Such a functional describes accurately linear dynamics but becomes irrelevant otherwise. Moreover, missing data are taken

into account during the computation: the method performs a smoothing of these aberrant values on the spatio-temporal neighborhood. Last, the time being viewed as an additional dimension, a consequence is the proportional increase of the problem's size. A huge memory is then requested for processing a large sequence.

A partial answer to the problem of dealing with complex dynamics and missing data is to include a model of the temporal evolution. This implies to be able to solve simultaneously the Image Model $\mathbb{I}$ and the evolution equation. This possibility is offered by the Data Assimilation framework described in the following Section.

3 The Data Assimilation framework

3.1 Mathematical setting

Let $\mathbf{X}$ being the state vector depending on the spatial coordinate $\mathbf{x}$ and time t. $\mathbf{X}$ is defined on $A = \Omega \times [0, T]$, Ω being the spatial domain and $[0, \mathbf{T}]$ the temporal domain.

We assume $\mathbf{X}$ is evolving in time according to:

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{x}, t) + \mathbb{M}(\mathbf{X})(\mathbf{x}, t) = \mathcal{E}_m(\mathbf{x}, t) \tag{5}$$

$\mathbb{M}$, named *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.

We consider having knowledge on the initial condition of the state vector at $t = 0$:

$$\mathbf{X}(\mathbf{x}, 0) = \mathbf{X}_b(\mathbf{x}) + \mathcal{E}_b(\mathbf{x}) \tag{6}$$

with $\mathbf{X}_b$ named the *background value* and $\mathcal{E}_b$ denotes the *background error*.

Observations $\mathbf{Y}(\mathbf{x}, t)$ are available at location $\mathbf{x}$ and date t and linked to the state vector through the observation equation:

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

Equation (7) 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 observations and the state vector. To be more general, the following will be used in this paper:

$$\mathbb{H}(\mathbf{Y}, \mathbf{X})(\mathbf{x}, t) = \mathcal{E}_O(\mathbf{x}, t) \tag{8}$$

which includes the previous formulation (7). The *observation error* $\mathcal{E}_O$ simultaneously represents the imperfection of the observation operator $\mathbb{H}$ and the measurement errors.

$\mathcal{E}_m$, $\mathcal{E}_b$ and $\mathcal{E}_O$ are assumed to be Gaussian and then fully characterized by their covariance matrices Q, B and R [21]. Let $\mathbf{Z}$ denote a Gaussian stochastic vector depending on a space-time coordinate $(\mathbf{x}, t)$; $\mathbf{Z} = \mathbf{Z}(x, t)$ and $\mathbf{Z}' = \mathbf{Z}(x', t')$ are random vectors on the two given locations. The covariance matrix Σ, computed for $\mathbf{Z}$ and $\mathbf{Z}'$, measures their dependency and is defined by:

$$\Sigma(\mathbf{x}, t, \mathbf{x}', t') = \int (\mathbf{Z} - \mathbb{E}\mathbf{Z})^T (\mathbf{Z}' - \mathbb{E}\mathbf{Z}') dP_{\mathbf{Z}, \mathbf{Z}'} \tag{9}$$

with $P_{\mathbf{Z}, \mathbf{Z}'}$ the joint distribution of $(\mathbf{Z}, \mathbf{Z}')$ and $\mathbb{E}$ denoting the expectation.

3.2 Variational formulation

In order to solve the system (5), (6) and (8) with respect to $\mathbf{X}$ having a maximal probability, the following functional, to be minimized, is defined:

$$
\begin{aligned}
E(\mathbf{X}) =& \frac{1}{2} \int_A \int_A \left(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) \right)^T (\mathbf{x}, t) Q^{-1}(\mathbf{x}, t, \mathbf{x}', t') \left(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) \right) (\mathbf{x}', t') d\mathbf{x} dt d\mathbf{x}' dt' \\
&+ \int_A \int_A \mathbb{H}(\mathbf{X}, \mathbf{Y})^T (\mathbf{x}, t) R^{-1}(\mathbf{x}, t, \mathbf{x}', t') \mathbb{H}(\mathbf{X}, \mathbf{Y})(\mathbf{x}', t') d\mathbf{x} dt d\mathbf{x}' dt' \\
&+ \int_\Omega \int_\Omega \left(\mathbf{X}(\mathbf{x}, 0) - \mathbf{X}_b(\mathbf{x}) \right)^T B^{-1}(\mathbf{x}, \mathbf{x}') \left(\mathbf{X}(\mathbf{x}', 0) - \mathbf{X}_b(\mathbf{x}') \right) d\mathbf{x} d\mathbf{x}'
\end{aligned}
\tag{10}
$$

As $\mathcal{E}_m$, $\mathcal{E}_b$ and $\mathcal{E}_O$ are assumed to be independent, the functional E represents the log-density of $\mathbf{X}$ law [2]. 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:

$$
\frac{\partial E}{\partial \mathbf{X}}(\eta) = \lim_{\gamma \to 0} \frac{d}{d\gamma} (E(\mathbf{X} + \gamma \eta))
\tag{11}
$$

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

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

We detail in Appendix A the determination of the Euler-Lagrange equation associated to (10). This leads to the following so-called Optimality System [9]:

$$
\lambda(\mathbf{x}, \mathbf{T}) = 0
\tag{13}
$$

$$
-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}} \right)^* \lambda = -\int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}} \right)^* (\mathbf{x}, t) R^{-1} \mathbb{H}(\mathbf{X}, \mathbf{Y})(\mathbf{x}', t') d\mathbf{x}' dt'
\tag{14}
$$

$$
\mathbf{X}(\mathbf{x}, 0) = \int_\Omega B\lambda(\mathbf{x}', 0) d\mathbf{x}' + \mathbf{X}_b(\mathbf{x})
\tag{15}
$$

$$
\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = \int_A Q\lambda(\mathbf{x}', t') d\mathbf{x}' dt'
\tag{16}
$$

Because the initial condition for λ is given at time T (13), λ must be computed backward in time using (14). Equation (14) makes use of two *adjoint operators* denoted $\left(\dfrac{\partial \mathbb{M}}{\partial \mathbf{X}} \right)^*$ and $\left(\dfrac{\partial \mathbb{H}}{\partial \mathbf{X}} \right)^*$. Adjoint operators are also called dual operators in the mathematics literature. For a given operator $\mathbb{K}$, we have:

$$
\int (\mathbb{K}(\eta))^T \lambda d\mu = \int \eta^T \mathbb{K}^*(\lambda) d\mu
\tag{17}
$$

for all integrable functions η and λ. 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 $[a, b]$:

$$\int_a^b \frac{\partial f}{\partial x}(x)g(x)dx = [f(x)g(x)]_{x=a}^{x=b} - \int_a^b f(x)\frac{\partial g}{\partial x}(x)dx$$

$$= \int_a^b f(x)\left(\left(\delta(x-b) - \delta(x-a)\right)g(x) - \frac{\partial g}{\partial x}(x)\right)dx$$

$$= \int_a^b f(x)\left(\frac{\partial g}{\partial x}\right)^*(x)dx$$

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

If $\mathbb{K}$ is a differential operator, the adjoint operator is then a compact notation for integration by parts.

3.3 Incremental algorithm

Solving the Optimality System is however not straightforward: the state vector is determined by equations (15,16) using the adjoint variable and the adjoint variable is determined by equations (13,14) using the state vector. To break this deadlock, an incremental method is applied. 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 called the *background variable* in the Data Assimilation literature, and $\delta\mathbf{X}$ is the incremental variable. $\mathbf{X}$ is then replaced by $\mathbf{X}_b + \delta\mathbf{X}$ in equations (14), (15) and (16). If $\mathbb{M}$ and $\mathbb{H}$ are non linear operators, we obtain:

$$\mathbb{M}(\mathbf{X}) \simeq \mathbb{M}(\mathbf{X}_b + \delta\mathbf{X}) = \mathbb{M}(\mathbf{X}_b) + \left.\frac{\partial\mathbb{M}}{\partial\mathbf{X}}\right|_{\mathbf{X}_b}(\delta\mathbf{X}) \tag{18}$$

$$\mathbb{H}(\mathbf{X}, \mathbf{Y}) \simeq \mathbb{H}(\mathbf{X}_b + \delta\mathbf{X}, \mathbf{Y}) = \mathbb{H}(\mathbf{X}_b, \mathbf{Y}) + \left.\frac{\partial\mathbb{H}}{\partial\mathbf{X}}\right|_{\mathbf{X}_b}(\delta\mathbf{X}) \tag{19}$$

from a first order Taylor development of $\mathbb{M}$ and $\mathbb{H}$ at $\mathbf{X}_b$. Equations (18) and (19) become equalities if $\mathbb{M}$ and $\mathbb{H}$ are linear. In both cases, (18,19) lead to the following new system:

$$\lambda(\mathbf{x}, \mathbf{T}) = 0 \tag{20}$$

$$-\frac{\partial\lambda}{\partial t} + \left(\left.\frac{\partial\mathbb{M}}{\partial\mathbf{X}}\right|_{\mathbf{X}_b}\right)^*\lambda = -\int_A \left(\left.\frac{\partial\mathbb{H}}{\partial\mathbf{X}}\right|_{\mathbf{X}_b}\right)^* R^{-1}\left(\mathbb{H}(\mathbf{X}_b, \mathbf{Y}) + \left.\frac{\partial\mathbb{H}}{\partial\mathbf{X}}\right|_{\mathbf{X}_b}(\delta\mathbf{X})\right)d\mathbf{x}'dt' \tag{21}$$

$$\mathbf{X}_b(\mathbf{x}, 0) = \mathbf{X}_b(\mathbf{x}) \tag{22}$$

$$\frac{\partial\mathbf{X}_b}{\partial t} + \mathbb{M}(\mathbf{X}_b) = 0 \tag{23}$$

$$\delta\mathbf{X}(\mathbf{x}, 0) = \int_\Omega B\lambda(\mathbf{x}', 0)d\mathbf{x}' \tag{24}$$

$$\frac{\partial\delta\mathbf{X}}{\partial t} + \left.\frac{\partial\mathbb{M}}{\partial\mathbf{X}}\right|_{\mathbf{X}_b}(\delta\mathbf{X}) = \int_A Q\lambda(\mathbf{x}', t')d\mathbf{x}'dt' \tag{25}$$

The background variable $\mathbf{X}_b$ is first calculated from equations (22) and (23). The adjoint variable λ is then obtained from the background variable using equations (20) and (21). Last, the incremental variable $\delta\mathbf{X}$ is obtained from the adjoint variable using equations (24) and (25).

If $\mathbb{M}$ and $\mathbb{H}$ are not linear, equations (20,21,24,25) only produce an approximated solution $\mathbf{X}_b + \delta\mathbf{X}$ due to the first order Taylor development described in (18) and (19). In this case, the incremental algorithm is applied iteratively until convergence. This iterative method, indexed by i, is summarized in the following:

1. Initialization ($i = 0$) :
   (a) Compute the background variable $\mathbf{X}_b$ from the initial condition of the state vector at $t = 0$ in (6):

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

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

   (b) Initialize the incremental variable:

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

2. Do $i = i + 1$ and :
   (a) Compute the adjoint variable λ from $t = T$ to $t = 0$:

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

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

   (b) Update the value of background variable:

   $$\mathbf{X}_b(\mathbf{x}, t) = \mathbf{X}_b(\mathbf{x}, t) + \delta\mathbf{X}(\mathbf{x}, t)$$

   (c) Compute the incremental variable $\delta\mathbf{X}$ (required for the next iteration) from $t = 0$ to $t = T$:

   $$\delta\mathbf{X}(\mathbf{x}, 0) = \int B(\mathbf{x}, \mathbf{x}')\lambda(\mathbf{x}', 0)d\mathbf{x}'$$

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

   Back to step 2 until $\|\delta\mathbf{X}\|^2 \leq \epsilon$.
3. Final result is $\mathbf{X}_b + \delta\mathbf{X}$.

4 Assimilation of images

This section explains how to solve the ill-posed Image Processing problems using the framework of Data Assimilation and constitutes the core of this research and one main contribution of the paper. Using Data Assimilation to solve Image Processing problems is a relatively recent domain. This idea has been described in [5] and the first application has been published in [17]. Studies have been done on curve tracking [16,17] and

determination of optical flow [14,15]. In [14] two evolution models are considered, a first one describing fluid motion by the transport of vorticity and a second one describing video motion using the transport of velocity by itself, stabilized by a diffusion process. In [15] a shallow water model is considered to describe the atmospheric circulation. In [7], a method is proposed to estimate the ocean surface circulation from SST data: images are assimilated in a dynamic image model to produce pseudo-observations for the oceanographic model. In this paper, we restrict to the case of ill-posed problems, which have been solved in the literature using Tikhonov regularization methods. We define a method to convert these problems, in a generic way, from the Tikhonov regularization to the Data Assimilation framework. The proposed method, which is the first innovation of the paper, rewrites the Image Model and the regularization term under the form of the system of three equations (5,6,8).

- First, the state and observation vectors have to be defined. Obviously, the observations will be images or processed images, but the components of the state vector will strongly depend on the studied problem. For example, segmentation, denoising and restoration use a state vector which is composed of the result's values for all pixels of the input image. Tracking, image registration and motion estimation use a state vector whose components are the values of the resulting vector field. Active contours use a curve.
- Second, a suitable equation describing the temporal evolution of the state vector has to be stated (Subsection 4.2). Next, an observation equation is written expressing the links between the state vector and the images (Subsection 4.3). Sometimes, this link is highly indirect. For instance, as it is not possible to deduce the ocean circulation from surface temperatures with a shallow-water model, a specific system has to be stated with relevant evolution and observation equations [7].
- Third, the initial condition should be defined.
- Last, the errors, characterized by their covariance matrices, must be chosen for the three equations. It is therefore crucial to understand the role of these matrices inside functional (10). The second innovation of the paper is therefore to conduct a full analysis of the errors and associated covariance matrices and their impact in term of spatio-temporal regularization (Subsection 4.1).

4.1 Covariance matrix

A covariance matrix Σ being defined by equation (9), its inverse is formally and implicitly defined [13] as:

$$\int \Sigma^{-1}(\mathbf{x}, \mathbf{x}'') \Sigma(\mathbf{x}'', \mathbf{x}') d\mathbf{x}'' = \delta(\mathbf{x} - \mathbf{x}') \tag{26}$$

We consider three possible choices of covariance and analyze their respective impact in a functional $\iint F^T(\mathbf{Z}) \Sigma^{-1} F(\mathbf{Z}) d\mathbf{x} d\mathbf{x}'$ which has to be minimized.

As a first example, let Σ be the Dirac covariance defined by $\Sigma(\mathbf{x}, \mathbf{x}') = \delta\left(\mathbf{x} - \mathbf{x}'\right)$. This covariance expresses a null interaction between two space locations. The Dirac function, $\delta\left(\right)$, has the following property:

$$\int_{\Omega} \delta\left(x'\right) \delta\left(x - x'\right) dx' = \delta\left(x\right) \tag{27}$$

By identification of (26) and (27), we have $\Sigma^{-1}(\mathbf{x}, \mathbf{x}') = \delta\left(\mathbf{x} - \mathbf{x}'\right)$ and:

$$\iint_{\Omega^2} F(\mathbf{Z})^T(\mathbf{x})\Sigma^{-1}(\mathbf{x}, \mathbf{x}')F(\mathbf{Z})(\mathbf{x}')d\mathbf{x}d\mathbf{x}' \qquad (28)$$
$$= \int_{\Omega} F(\mathbf{Z})^T(\mathbf{x})F(\mathbf{Z})(\mathbf{x})d\mathbf{x} = \int_{\Omega} \|F(\mathbf{Z})\|^2 \, d\mathbf{x}$$

A Dirac covariance is therefore acting like a zero-order regularization on $F(\mathbf{Z})$.

More generally, we consider the case of an isotropic covariance which is written as $\Sigma(\mathbf{x}, \mathbf{x}') = s(\mathbf{x} - \mathbf{x}')$. In this case, determining its inverse from (26) is equivalent to determine its inverse convolution defined by:

$$s^{-1} \star s(\mathbf{x}) = \delta(\mathbf{x}) \qquad (29)$$

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

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

Using the convolution theorem, we have:

$$\widehat{s^{-1}}(\omega)\widehat{s}(\omega) = 1$$
$$\widehat{s^{-1}}(\omega) = \frac{1}{\widehat{s}(\omega)}$$

and the inverse convolution $s^{-1}(x)$ is obtained by the inverse Fourier transform of $\frac{1}{\widehat{s}}$.

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

$$\widehat{s^{-1}}(\omega) = \frac{1 + \sigma^2\omega^2}{2\sigma}$$
$$s^{-1}(\mathbf{x}) = \frac{1}{2\sigma}\left(\delta\left(\mathbf{x}\right) - \sigma^2\delta''(\mathbf{x})\right)$$

We replace the expression of $\Sigma^{-1}(\mathbf{x}, \mathbf{x}')$ by $s^{-1}(\mathbf{x} - \mathbf{x}')$ in the functional (28):

$$\iint_{\Omega^2} F(\mathbf{Z})^T(\mathbf{x})\Sigma^{-1}(\mathbf{x}, \mathbf{x}')F(\mathbf{Z})(\mathbf{x}')d\mathbf{x}d\mathbf{x}'$$
$$= \frac{1}{2\sigma}\int_{\Omega} F(\mathbf{Z})^T(\mathbf{x})\left(F(\mathbf{Z}) - \sigma^2\frac{\partial^2 F(\mathbf{Z})}{\partial\mathbf{x}^2}\right)d\mathbf{x}$$
$$= \frac{1}{2\sigma}\int_{\Omega}\left(\|F(\mathbf{Z})\|^2 + \sigma^2\left\|\frac{\partial F(\mathbf{Z})}{\partial\mathbf{x}}\right\|^2\right)d\mathbf{x} \qquad (30)$$

Integration by parts is applied for deriving (30), assuming null boundary terms. The exponential covariance is thus associated to a first-order regularization of $F(\mathbf{Z})$.

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

is thus $\widehat{s^{-1}}(\omega) = \dfrac{1}{\sigma\sqrt{\pi}} \exp\left(\dfrac{\sigma^2}{4}\omega^2\right)$, but its inverse Fourier transform can not be directly computed. Let us replace the exponential by its infinite series:

$$\widehat{s^{-1}}(\omega) = \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} \frac{1}{n!} \left(\frac{\sigma^2}{4}\omega^2\right)^n$$
$$= \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} a_n \omega^{2n} \tag{31}$$

with $a_n = \dfrac{1}{n!} \left(\dfrac{\sigma}{2}\right)^{2n}$. Remembering that 1 is the inverse Fourier transform of the Dirac function and using the theorem of Fourier derivation, equation (31) leads to:

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

Let us examine the impact of such covariance in the functional (28):

$$\iint F(\mathbf{Z})^T(\mathbf{x}) \Sigma^{-1}(\mathbf{x},\mathbf{x}') F(\mathbf{Z})(\mathbf{x}') d\mathbf{x} d\mathbf{x}'$$
$$= \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} (-1)^n a_n \int F(\mathbf{Z})^T(\mathbf{x}) \left(\int \delta^{(2n)}(\mathbf{x}-\mathbf{x}') F(\mathbf{Z})(\mathbf{x}') d\mathbf{x}' \right) d\mathbf{x}$$
$$= \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} (-1)^n a_n \int F(\mathbf{Z})^T(\mathbf{x}) F^{(2n)}(\mathbf{Z})(\mathbf{x}) d\mathbf{x}$$
$$= \frac{1}{\sigma\sqrt{\pi}} \int \sum_{n\geq 0} a_n \left\| \frac{\partial^n F(\mathbf{Z})}{\partial \mathbf{x}^n} \right\|^2 (\mathbf{x}) d\mathbf{x}$$

Again, assuming null boundary conditions, we use a series of integration by parts to derive to the final result. The Gaussian exponential covariance then corresponds to the Tikhonov regularization at any order of $F(\mathbf{Z})$.

The inversion of a covariance matrix Σ 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 it using numerical techniques. Unfortunately, if the discrete matrix is large, the inversion is costly and often numerically unstable. For further details, the reader is referred to [13,21].

4.2 The evolution model

Tikhonov regularization is in fact assuming spatial properties of the result from heuristics on the dynamics. These heuristics, with a better knowledge on the image dynamics, could be efficiently written as an evolution equation of the state vector $\mathbf{X}$. This evolution law imposes a temporal regularity of $\mathbf{X}$ and consequently impacts in specific spatial configurations.

A first and simple heuristic about dynamics is to assume $\mathbf{X}$ being constant over time, which is expressed by:

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

or:

$$\frac{\partial \mathbf{X}}{\partial t} + \frac{\partial \mathbf{X}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial t} = 0 \tag{32}$$

This is a transport equation as $\frac{\partial \mathbf{x}}{\partial t}$ is a velocity vector. Identifying equation (32) and (5), 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 (32) as evolution equation is given in Section 5 for optical flow estimation.

Another possible heuristic is to express the transport of the state vector as a diffusion process, a physical law applying for chemical species or temperature. The general formulation is: $\frac{\partial \mathbf{X}}{\partial t} = \nabla^T(D\nabla\mathbf{X})$ and by identification with equation (5), 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 [27]. It is also possible to drive the diffusion according to image characteristics. A standard example is the Perona & Malik diffusion [18]: 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 D can also take into account the orientation of the image gradient [24, 25] resulting in spatial regularization properties similar to that of Nagel [11].

The two previous definitions of $\mathbb{M}$ remain restrictive for image sequences displaying complex dynamics. In these cases, the evolution model could be built using prior information on the observed data. As a first example, dynamics can be approximated by piecewise linear functions whose parameters are estimated by analyzing the evolution of the image probability density function with a particule filter method. However, this issue remains complex and widely open. As a second example, an evolution model based on a physical law is considered. In [7,15] a shallow-water equation is chosen for modeling the evolution of apparent motion, which 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.

The covariance matrix Q, associated to the model error $\mathcal{E}_m$, is used in functional (10) for regularizing $\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})$. For this reason Q is chosen as Dirac covariance as explained in Subsection 4.1. In such a way, we are ensuring that the temporal evolution of $\mathbf{X}$ is coherent with the heuristics involved in the model $\mathbb{M}$.

4.3 The observation equation

As previously pointed out, the observation equation describes the links between the state vector and the observations. In the standard framework of Image Processing, an image or a sequence of images provides the observations in the Image Model $\mathbb{I}$ expressed in equation (1). The operator $\mathbb{H}$, as it appears in equation (8), is then defined as the image operator $\mathbb{I}$ i.e. $\mathbb{H}(\mathbf{X}, \mathbf{Y}) \equiv \mathbb{I}(\mathbf{X}, \mathbf{Y})$.

The observation error $\mathcal{E}_{\mathbf{Y}}$, characterized by its covariance matrix R, has also to be specified. R weights the contribution of observations in equation (21). Its inverse

should then have values close to zero when observations should be discarded, which is the case for missing data. Using the Dirac covariance, 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 whose size corresponds to the number of components of $\mathbb{H}$ and the inverse is:

$$R^{-1}(\mathbf{x}, t, \mathbf{x}', t') = \delta\left(\mathbf{x} - \mathbf{x}'\right)\delta\left(t - t'\right)r^{-1}(\mathbf{x}, t) \tag{33}$$

The matrix 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 be included in the computation of the solution. Assuming the availability of a function f measuring the confidence in observation data ($f \in [0, 1]$, $f = 0$ for no confidence), one possible formulation of r^{-1} is:

$$r^{-1}(\mathbf{x}, t) = r_0(1 - f(\mathbf{x}, t)) + r_1 f(\mathbf{x}, t) \tag{34}$$

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

$$f(\mathbf{x}, t) = f_{\text{sensor}}(\mathbf{x}, t)f_{\text{noise}}(\mathbf{x}, t)f_{\mathbb{H}}(\mathbf{x}, t) \tag{35}$$

f_{sensor} indicates the availability of observation values: it is equal to 0 if 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.

With such definition of R, observation values with a low confidence will not be considered during the computation of the solution of the Image Processing problem.

4.4 Conclusion

In this Section, we considered the class of ill-posed Image Processing problems which are usually solved by constraining spatially and/or temporally the solution. We proposed a generic method to convert the Image Model and the spatio-temporal regularity constraint on the result in term of Data Assimilation components, as expressed in equations (5,6,8). The Image Model is taken as 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 observations can be evaluated. The evolution equation describes the temporal evolution of the state vector and we proposed two generic choices for the operator $\mathbb{M}$. However, both are too restrictive to deal with complex temporal dynamics and prior knowledge should be included when available. The coherency of the temporal evolution through the operator $\mathbb{M}$ is ensured by the covariance matrix Q and we gave three examples of regularization. In the next Section, we illustrate 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 t and $t + \triangle t$, verifying:

$$I(\mathbf{x} + \mathbf{W}(\mathbf{x}, t)\triangle t, t + \triangle t) = I(\mathbf{x}, t) \tag{36}$$

As this equation is non linear with respect to $\mathbf{W}$, the left member of equation (36) is often linearized using a first order Taylor development around $\triangle t = 0$. This provides the so-called optical flow constraint equation [6]:

$$\nabla I^T(\mathbf{x}, t)\mathbf{W}(\mathbf{x}, t) + \frac{\partial I}{\partial t}(\mathbf{x}, t) = 0 \quad \forall \mathbf{x} \in \Omega \tag{37}$$

By identifying (37) with (1), we have $\mathbf{X} = \mathbf{W}$, $\mathbf{Y} = (\nabla I, I_t)$ and $\mathbb{I} = \nabla I^T(\mathbf{x}, t)\mathbf{W}(\mathbf{x}, t) + \frac{\partial I}{\partial t}(\mathbf{x}, t)$. 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 [6] or spatially and temporally [26] the solution. Equation (37) is often preferred to (36) because it is linear and the associated Euler-Lagrange equation can be discretized using an explicit and robust numerical scheme. However, it is possible to directly use the equation (36) for estimating the optical flow. This has been described in [3] where the resulting Euler-Lagrange equation has been approximated with a semi-implicit scheme. The advantage of equation (36) is that $\mathbf{W}$ can be estimated even for high velocity norms. Equation (37) is an approximation and only well suited for low velocity except if incremental algorithms [12,19] or scale-space methods [1] are considered.

In this paper, we choose however to consider the optical flow constraint (37) in order to illustrate the tasks to be applied for going from an ill-posed Image Processing problem to a Data Assimilation system, compare it with state-of-the-art methods, and prove the advantage of Data Assimilation when processing noisy acquisitions including missing data.

5.1 Observation and evolution models

The optical flow constraint is chosen as image model. As $\mathbb{H} \equiv \mathbb{I}$, the observation model is:

$$\mathbb{H}(\mathbf{W}, I)(\mathbf{x}, t) = \nabla I(\mathbf{x}, t)^T\mathbf{W}(\mathbf{x}, t) + I_t(\mathbf{x}, t) \tag{38}$$

With this choice, the field $\mathbf{W}(\mathbf{x}, t)$ of velocity vectors is now considered as the state vector $\mathbf{X}(\mathbf{x}, t)$ and the image gradients $(\nabla I(\mathbf{x}, t), I_t(\mathbf{x}, t))$ constitutes the observation vector $\mathbf{Y}(\mathbf{x}, t)$.

We have to define an appropriate observation error. Equations (33) and (34) are used to define the inverse of R and to locate the observation values which must be discarded. The observation model $\mathbb{H}$ 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 (35) is used as the observation confidence. Without any information, f_{noise} is assumed to be equal to 1. f_{sensor} is set to 0 if 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 (37) is then degenerated on pixels belonging to them. For avoiding further considering these points, $f_\mathbb{H}$ is defined by:

$$f_\mathbb{H}(\mathbf{x}, t) = 1 - \exp(-\|\nabla_3 I(\mathbf{x}, t)\|^2) \tag{39}$$

where ∇_3 denotes the spatio-temporal gradient operator.

The transport of the velocity, equation (32), is taken as evolution equation:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \mathbf{W}^T \mathbf{W} = 0 \tag{40}$$

This equation is rewritten as a two-component system:

$$\frac{\partial U}{\partial t} + U U_x + V U_y = 0 \tag{41}$$

$$\frac{\partial V}{\partial t} + U V_x + V V_y = 0 \tag{42}$$

and by identification of equation (5) with (40) and (41,42), the evolution model is:

$$\mathbb{M}(\mathbf{W}) = \begin{pmatrix} \mathbb{M}_1(\mathbf{W}) & \mathbb{M}_2(\mathbf{W}) \end{pmatrix}^T$$
$$= \begin{pmatrix} U U_x + V U_y & U V_x + V V_y \end{pmatrix}^T$$

$\mathbf{W}$ being a two-component vector, the Q matrix is of size 2×2 and chosen as:

$$Q(\mathbf{x}, t, \mathbf{x}', t') = q(\mathbf{x} - \mathbf{x}', t - t') \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{43}$$

with $q(\mathbf{x}, t) = \exp(-\frac{1}{\sigma}(\|\mathbf{x}\| + |t|))$ or $q(\mathbf{x}, t) = \delta(\mathbf{x})\delta(t)$. We have tested both covariance matrices with $\sigma = 1$ without noticing significant differences on the results.

The background value at $t = 0$ (equation (6)) has also to be provided: we make use of Horn and Schunck's algorithm [6] to compute the velocity field on the two first frames of the sequence. We consider the background error $B(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$ for measuring the quadratic discrepancy between $\mathbf{X}_b(\mathbf{x})$ and $\mathbf{X}(\mathbf{x}, 0)$.

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 (11), we obtain:

$$\frac{\partial \mathbb{M}_1}{\partial \mathbf{W}}(\eta) = \frac{\partial \mathbb{M}_1}{\partial U}(\eta^1) + \frac{\partial \mathbb{M}_1}{\partial V}(\eta^2) = U \eta_x^1 + U_x \eta^1 + V \eta_y^1 + U_y \eta^2$$
$$\frac{\partial \mathbb{M}_2}{\partial \mathbf{W}}(\eta) = \frac{\partial \mathbb{M}_2}{\partial U}(\eta^1) + \frac{\partial \mathbb{M}_2}{\partial V}(\eta^2) = U \eta_x^2 + V_y \eta^2 + V \eta_y^2 + V_x \eta^1$$

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 definition (17),

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

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

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

with $\lambda = \left(\lambda^1 \ \lambda^2\right)^T$. Details are reported in Appendix B.2. In a compact form, the adjoint operator of $\mathbb{M}$ is written:

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

$$= -\left(\nabla\lambda_1 \ \nabla\lambda_2\right)^T \mathbf{W} - \left(\nabla^\perp V \ -\nabla^\perp U\right)^T \lambda$$

with $\nabla^\perp U = \left(U_y \ -U_x\right)^T$.

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, differential and adjoint operators values calculated in Subsection 5.2, the three PDEs (23,21,25) become:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla\mathbf{W}^T\mathbf{W} = 0 \tag{44}$$

$$-\frac{\partial \lambda}{\partial t} - \nabla\lambda^T\mathbf{W} - (\nabla^\perp\mathbf{W})^T\lambda = -\nabla I R^{-1} \star L \tag{45}$$

$$\frac{\partial \delta\mathbf{W}}{\partial t} + \nabla\delta\mathbf{W}^T\mathbf{W} + \nabla\mathbf{W}^T\delta\mathbf{W} = Q \star \lambda \tag{46}$$

with $\nabla^\perp\mathbf{W} = \left(\nabla^\perp V \ -\nabla^\perp U\right)$, and $L = I_t + \nabla I^T(\mathbf{W} + \delta\mathbf{W})$. The covariance matrices Q and R being chosen isotropic, they only depend on $\mathbf{x} - \mathbf{x}'$ and $t - t'$, and the right members of equations (45) and (46) can then be expressed as a convolution product. For evaluating these right members at a given date, the knowledge of L and λ over the whole temporal domain is required. However, choosing a Dirac matrix for Q and R^{-1} reduces the convolution product to a simple multiplication. In (45) computation of $\lambda(t)$ is only depending on values at $(t+1)$ and in (46) $\delta\mathbf{X}(t)$ is obtained from variables at $(t-1)$. The algorithm becomes a frame-by-frame process: the whole sequence of state and observation vectors, adjoint and incremental variables may be left on the mass storage, excepted frames implied in the computation.

The three equations are discretized using a finite difference technique. Let us first examine equation (44): 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 known as being 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 scheme, a diffusive term is often 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 [15]. This is known under the name of Lax-Wendroff method. Such an equation can then be approximated using an explicit Euler scheme (with Courant-Friedrich-Levy condition) or an implicit Euler scheme. This has the drawback of smoothing the solution and not preserving sharp discontinuities. We propose, in the following, a stable scheme for the advection equation, without adding this diffusive term, by using a splitting method [23]. As $\mathbf{W}(\mathbf{x}, t)$ is a vector of $\mathbb{R}^2$, equation (44) has two components (41) and (42). The first one combines a term of linear advection in direction y and non linear one in direction x and is expressed as a two-equation system using the splitting method:

$$\frac{\partial U}{\partial t} + U U_x = 0 \qquad (47)$$

$$\frac{\partial U}{\partial t} + V U_y = 0 \qquad (48)$$

Equation (47) is rewritten with the Lax-Friedrich method [20] as $\dfrac{\partial U}{\partial t} + \dfrac{\partial F(U)}{\partial x} = 0$ with $F(U) = \dfrac{1}{2} U^2$. This new equation is discretized by:

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

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 while $\triangle t$ satisfies the Courant-Friedrich-Levy condition. The linear advection (48) is discretized using an explicit shock scheme [20]:

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

In the same way, it can be seen that the second component of (44) 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 (45) combines a linear advection ($\nabla \lambda^T \mathbf{W}$), a term of reaction $((\nabla^\perp \mathbf{W})^T \lambda)$ and a forcing term ($\nabla I R^{-1} \star L$). Its first component is $-\dfrac{\partial \lambda^1}{\partial t} - U \lambda_x^1 - V_y \lambda^1 - V \lambda_y^1 + V_x \lambda^2 = \dfrac{\partial I}{\partial x} A$ with $A = -R^{-1} \star L$. It is split into two parts. The first part contains the linear advection in direction x and the reaction term: $-\dfrac{\partial \lambda^1}{\partial t} - U \lambda_x^1 - V_y \lambda^1 = 0$ and is discretized in the same way as (48) with an explicit shock scheme. However, the equation is retrograde and its initial condition is given at time T:

$$(\lambda^1)_{i,j}^{k-1} = \left(1 + \frac{\triangle t}{2}(V_{i,j+1}^k - V_{i,j-1}^k) \right)(\lambda^1)_{i,j}^k +$$

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

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

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

Having the same structure, the second component of (45) is discretized with the same method. The complete numerical scheme is described in Appendix C.2.

The last equation, (46), is similar to equation (45): a linear advection with a reaction term and a forcing term. We therefore use the same discretization technique. The full numerical scheme is detailed in Appendix C.3.

5.4 Results

The "taxi" sequence and a synthetic sequence have been chosen for discussing results. In both cases, image gradients are computed with a convolution method and a derivative Gaussian kernel whose variance is set to 1. The incremental algorithm (Subsection 3.3) is iterated 5 times.

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 and analysis purposes of our method. Figures 1 to 3 show the results obtained on three frames of the sequence. These results are qualitatively similar, illustrating that both methods are equivalent when observation values are available on the whole sequence. Having chosen Horn & Schunck or another image processing method does not really matter, because we are not interested in discussing a quantitative comparison of optical flow methods but in proving the efficiency of Data Assimilation for dealing with missing data and complex dynamics. Consequently, a second experiment is designed for analyzing the issue of missing data. A large region around the white car, denoted $\Re$, is set to zero (black rectangle) on one frame of the sequence to simulate a sensor failure. To indicate 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 and to the evolution equation. A similar experiment is performed by setting several small regions to zero on one frame and flagged them as not acquired with f_{sensor} equal to zero. 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 approximates the temporal dynamics of these data and compensates the missing acquisition.
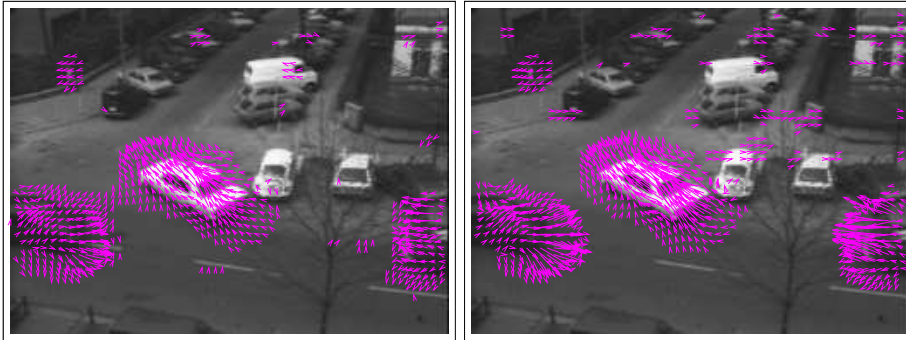

Fig. 1 Comparison Data Assimilation (left) / Horn-Schunk (right) – frame 3.
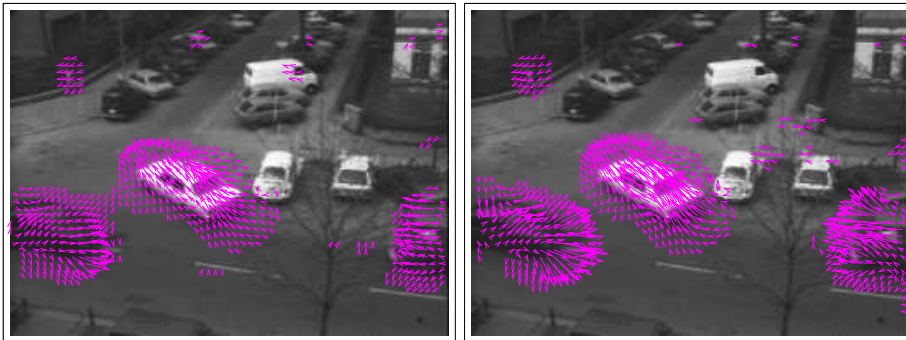

Fig. 2 Comparison Data Assimilation (left) / Horn-Schunk (right) – frame 6.

A third experiment is dedicated to prove that using the real dynamics within the evolution model ensures getting correct estimation of motion. For that purpose, we built a synthetic sequence displaying one square moving horizontally from the left up part of the image and one moving vertically from the right bottom part. At the end of the sequence, the two squares meet each other. Figures 7(b), 8(b) and 9(b) show the results with Horn & Schunk's method which fails to estimate a correct velocity direction when the squares meet. This is due to an over-regularization by the cost function. Figures 7(a), 8(a) and 9(a) show the results with Data Assimilation: motion is better estimated because the evolution model correctly describes the dynamics and avoids the negative effects of spatial regularization.

6 Conclusion

In this paper we proposed a general framework to solve ill-posed Image Processing problems by Data Assimilation methods. This is an alternative to the space-time ap-
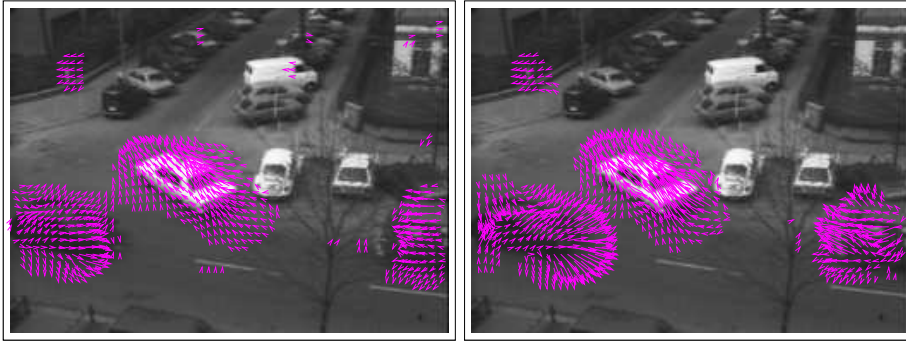
Fig. 3 Comparison Data Assimilation (left) / Horn-Schunk (right) – frame 9.
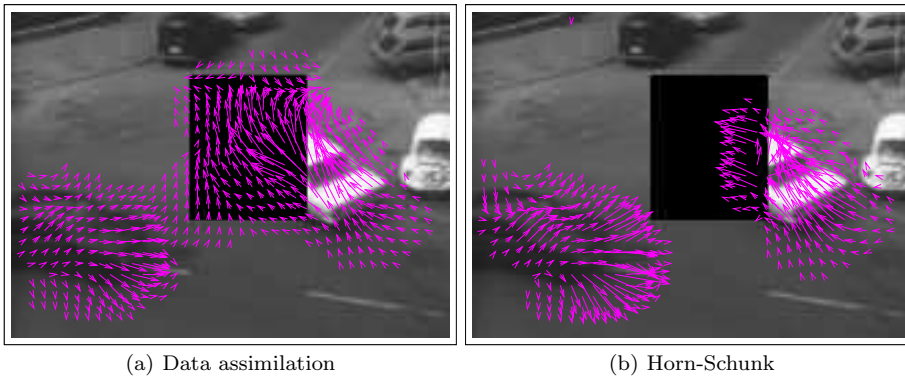

(a) Data assimilation   (b) Horn-Schunk

Fig. 4 Missing data on a large region in frame 5.

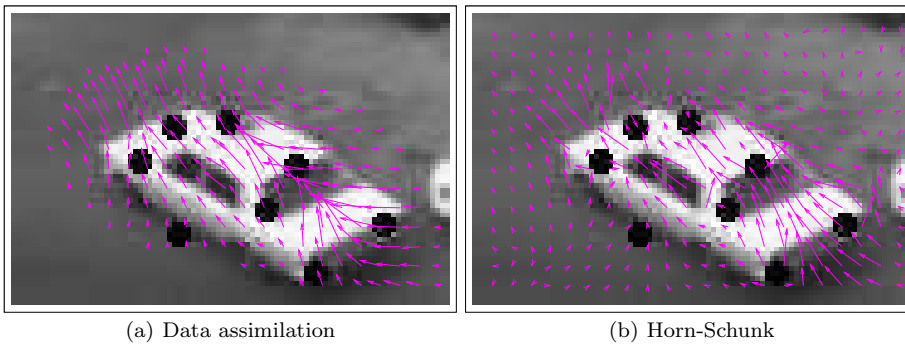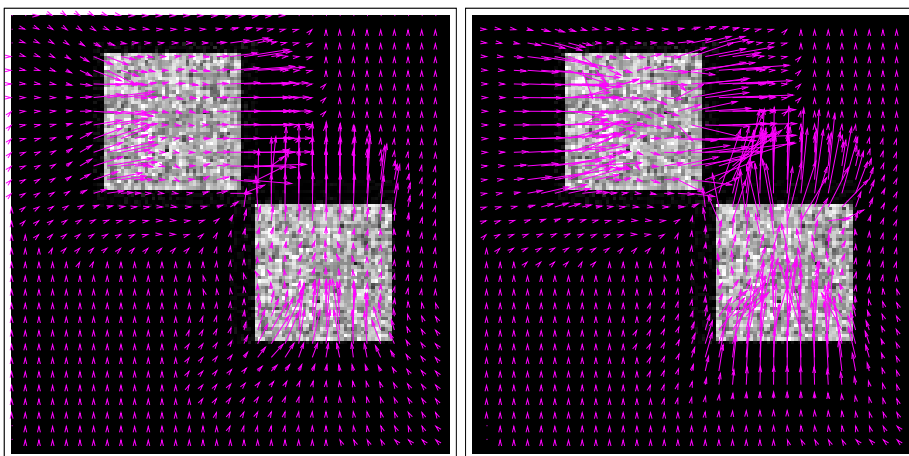(a) Data assimilation   (b) Horn-Schunk

Fig. 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

Fig. 6 Missing data on the whole frame 5.

(a) Data Assimilation

(b) Horn-Schunk

Fig. 7 Results on synthetic sequence - frame 4.

proach, described in the introduction, which constrains the solution's variations in space and time.

If the dynamics is approximately known and expressed as an evolution equation, we show how this information is used, simultaneously with the observation equation, in the framework of Data Assimilation, to temporally constrain the solution and obtain a better result. We extensively describe how to formalize the image processing problem using the Data Assimilation framework.

The impact of covariance matrices on the energy minimization has been investigated. Their regularization properties have been described for three different cases. The knowledge of the temporal dynamics and the choice of relevant covariance matrices make it possible to handle the problem of missing and noisy data. For that purpose, the observation error is described by a specific covariance matrix, with high values on pixels corresponding to missing and noisy data, which are then discarded during the com-
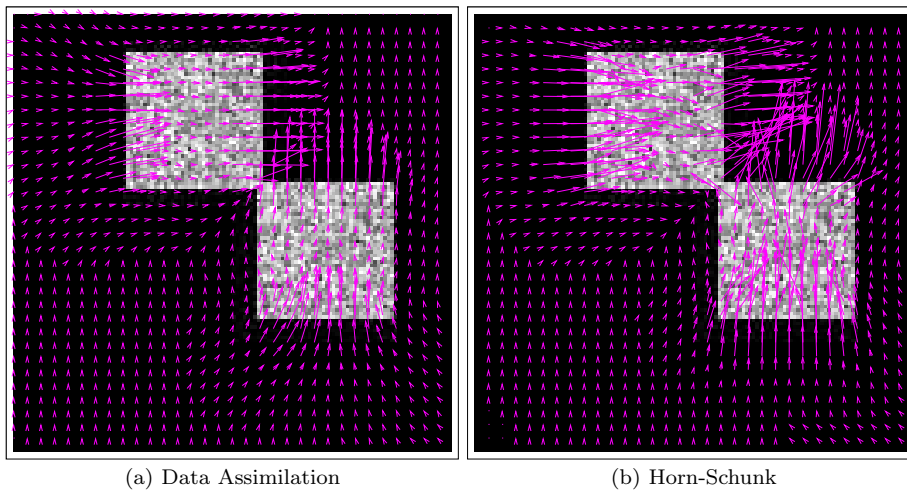
(a) Data Assimilation   (b) Horn-Schunk

Fig. 8 Results on synthetic sequence - frame 7.

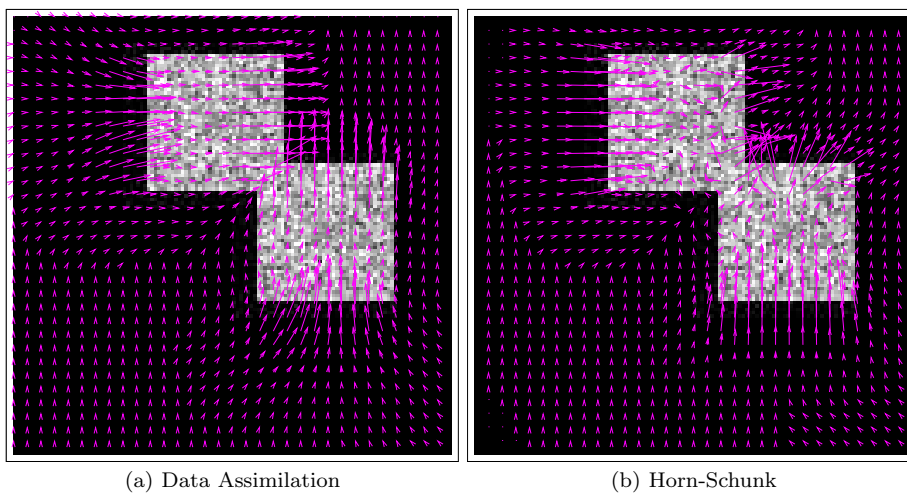(a) Data Assimilation   (b) Horn-Schunk

Fig. 9 Results on synthetic sequence - frame 10.

putation 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.

Due to specific choices of covariances, the 4D-Var algorithm described in the paper allows a local computation of the state vector, on a frame-by-frame basis. It therefore has memory management advantages compared to the space-time approach, which requires to include the whole sequence in memory, as the linear system is solved in the space-time domain.

The general approach has been illustrated on estimation of the optical flow by assimilating image gradient observations in a model, that describes the evolution of

velocity by itself. This transport equation is simple and correctly approximates the dynamics, which is locally translational and uniform. The chosen observation equation is the optical flow constraint, linearly linking velocity and image brightness gradients. The discretization leads to an unstable numerical scheme if standard finite difference methods are used. To overcome this problem, a robust scheme, based on a splitting method, has been proposed in the paper.

A first perspective of this work is to investigate the evolution equation issue. For some applications it is possible to use a physically consistent equation: 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 motion. A solution is to build a parametric model of the image dynamics: the spatio-temporal domain is first divided in sub-domains and the transport equation is assumed to correctly describe the dynamics on each of them.

A second perspective is to investigate the spatial regularization of the state vector $\mathbf{X}$ through the observation covariance matrix R.

A last perspective addresses the experimental application chosen in the paper: the estimation of optical flow. The optical flow equation has been used as observation equation. This equation has the advantage to be linear but 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 is supposed to be differentiable. The 4D-Var algorithm, described in Subsection 3.3, could then be used with such observation equation.

A The Euler-Lagrange equation of E (10)

Let us first compute the derivative of E in direction η using definition (11):

$$
\begin{aligned}
E(\mathbf{X} + \gamma\eta) = {} & \frac{1}{2} \int_A \int_A \left(\mathbf{X}_t + \gamma\eta_t + \mathbb{M}(\mathbf{X} + \gamma\eta)\right)^T Q^{-1} \left(\mathbf{X}_t + \gamma\eta_t + \mathbb{M}(\mathbf{X} + \gamma\eta)\right)^T d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_A \int_A \mathbb{H}(\mathbf{X} + \gamma\eta, \mathbf{Y})^T R^{-1} \mathbb{H}(\mathbf{X} + \gamma\eta, \mathbf{Y}) d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_\Omega \int_\Omega (\mathbf{X} + \gamma\eta - \mathbf{X}_b)^T B^{-1} (\mathbf{X} + \gamma\eta - \mathbf{X}_b) d\mathbf{x}d\mathbf{x}' \\
\frac{d}{d\gamma} E(\mathbf{X} + \gamma\eta) = {} & \int_A \int_A \left(\eta_t + \frac{d}{d\gamma}\mathbb{M}(\mathbf{X} + \gamma\eta)\right)^T Q^{-1} \left(\mathbf{X}_t + \gamma\eta_t + \mathbb{M}(\mathbf{X} + \gamma\eta)\right) d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_A \int_A \frac{d}{d\gamma} \left(\mathbb{H}(\mathbf{X} + \gamma\eta, \mathbf{Y})^T\right) R^{-1} \mathbb{H}(\mathbf{X} + \gamma\eta, \mathbf{Y}) d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_\Omega \int_\Omega \eta^T B^{-1} (\mathbf{X} + \gamma\eta - \mathbf{X}_b) d\mathbf{x}d\mathbf{x}'
\end{aligned}
$$

Let us γ tend to zero:

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

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

$$\frac{\partial E}{\partial \mathbf{X}}(\eta) = \int_A \int_A \eta^T \left(\delta(t = \mathbf{T}) - \delta(t = 0) - \frac{\partial}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* \right) Q^{-1} \left(\mathbf{X}_t + \mathbb{M}(\mathbf{X}) \right) d\mathbf{x} dt d\mathbf{x}' dt'$$

$$+ \int_A \int_A \eta^T \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1} \mathbb{H}(\mathbf{X}, \mathbf{Y}) d\mathbf{x} dt d\mathbf{x}' dt'$$

$$+ \int_\Omega \int_\Omega \eta^T B^{-1} (\mathbf{X} - \mathbf{X}_b) d\mathbf{x} d\mathbf{x}'$$

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

$$\frac{\partial E}{\partial \mathbf{X}}(\eta) = \int_\Omega \eta^T(\mathbf{x}, \mathbf{T}) \lambda(\mathbf{x}, \mathbf{T}) d\mathbf{x} - \int_\Omega \eta^T(\mathbf{x}, 0) \lambda(\mathbf{x}, 0) d\mathbf{x}$$

$$+ \int_A \eta^T(\mathbf{x}, t) \left(-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* (\lambda) \right) d\mathbf{x} dt$$

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

$$+ \int_\Omega \eta^T(\mathbf{x}, 0) \left(\int_\Omega B^{-1} (\mathbf{X}(\mathbf{x}', 0) - \mathbf{X}_b(\mathbf{x}')) d\mathbf{x}' \right) d\mathbf{x}$$

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

$$\lambda(\mathbf{x}, \mathbf{T}) = 0 \tag{49}$$

$$-\lambda(\mathbf{x}, 0) + \int_\Omega B^{-1} (\mathbf{X}(\mathbf{x}', 0) - \mathbf{X}_b(\mathbf{x}')) d\mathbf{x}' = 0 \tag{50}$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* (\lambda) + \int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1} \mathbb{H}(\mathbf{X}, \mathbf{Y}) d\mathbf{x}' dt' = 0 \tag{51}$$

Using the definition of inverse covariance (26), Equation (50) is rewritten as:

$$\mathbf{X}(\mathbf{x}, 0) = \mathbf{X}_b(\mathbf{x}) + \int_\Omega B(\mathbf{x}, \mathbf{x}') \lambda(\mathbf{x}', 0) d\mathbf{x}' \tag{52}$$

and the state vector is expressed from λ using (12):

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = \int_A Q(\mathbf{x}, t, \mathbf{x}', t') \lambda(\mathbf{x}', t') d\mathbf{x}' dt' \tag{53}$$

Equations (49), (51), (52) and (53) are 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:

$$\frac{\partial \mathbb{M}}{\partial \mathbf{W}} = \begin{pmatrix} \dfrac{\partial \mathbb{M}_1}{\partial U} & \dfrac{\partial \mathbb{M}_1}{\partial V} \\ \dfrac{\partial \mathbb{M}_2}{\partial U} & \dfrac{\partial \mathbb{M}_2}{\partial V} \end{pmatrix} \tag{54}$$

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:

$$\frac{\partial\mathbb{M}_1}{\partial U}^T\eta^1 = U\eta_x^1 + V\eta_y^1 + U_x\eta^1$$

$$\frac{\partial\mathbb{M}_1}{\partial V}^T\eta^2 = U_y\eta^2$$

$$\frac{\partial\mathbb{M}_2}{\partial U}^T\eta^1 = V_x\eta^1$$

$$\frac{\partial\mathbb{M}_2}{\partial V}^T\eta^2 = U\eta_x^2 + V\eta_y^2 + V_y\eta^2$$

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

In equation (25), 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 (54)) that the Jacobian matrix does not need to be transposed to compute the derivative in a given direction. It is therefore evaluated as:

$$\frac{\partial\mathbb{M}}{\partial\mathbf{W}}(\delta\mathbf{W}) = \frac{\partial\mathbb{M}}{\partial\mathbf{W}}\delta\mathbf{W} = \begin{pmatrix} \dfrac{\partial\mathbb{M}_1}{\partial U}\delta U + \dfrac{\partial\mathbb{M}_1}{\partial V}\delta V \\ \dfrac{\partial\mathbb{M}_2}{\partial U}\delta U + \dfrac{\partial\mathbb{M}_2}{\partial V}\delta V \end{pmatrix}$$

$$= \begin{pmatrix} U\delta U_x + V\delta U_y + U_x\delta U + U_y\delta V \\ U\delta V_x + V\delta V_y + V_x\delta U + V_y\delta V \end{pmatrix}$$

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

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

$$\left\langle \frac{\partial\mathbb{M}}{\partial\mathbf{W}}(\delta\mathbf{W}),\lambda \right\rangle_{L^2} = \left\langle \delta\mathbf{W},\left(\frac{\partial\mathbb{M}}{\partial\mathbf{W}}\right)^*(\lambda) \right\rangle_{L^2}$$

$$= \int\left(\frac{\partial\mathbb{M}}{\partial\mathbf{W}}(\delta\mathbf{W})\right)^T\lambda d\mathbf{x}$$

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

$$\int\left(\frac{\partial\mathbb{M}}{\partial\mathbf{W}}(\delta\mathbf{W})\right)^T\lambda d\mathbf{x} = \int\begin{pmatrix} U\delta U_x + V\delta U_y + U_x\delta U + U_y\delta V \\ U\delta V_x + V\delta V_y + V_x\delta U + V_y\delta V \end{pmatrix}^T\begin{pmatrix}\lambda^1\\\lambda^2\end{pmatrix}d\mathbf{x}$$

$$= \int\Big(-\delta U\partial_x(U\lambda^1) - \delta U\partial_y(V\lambda^1) + \delta U U_x\lambda^1 + \delta V U_y\lambda^1$$

$$\delta U V_x\lambda^2 - \delta V\partial_x(U\lambda^2) - \delta V\partial_y(V\lambda^2) + \delta V V_y\lambda^2\Big)d\mathbf{x}$$

$$= \int\left(\delta U\ \delta V\right)\begin{pmatrix}-U\lambda_x^1 - V_y\lambda^1 - V\lambda_y^1 + V_x\lambda^2 \\ -U_x\lambda^2 - U\lambda_x^2 - V\lambda_y^2 + U_y\lambda^1\end{pmatrix}d\mathbf{x}$$

$$= \int\left(\delta U\ \delta V\right)\begin{pmatrix}-U\partial_x - V\partial_y - V_y & V_x \\ U_y & -U\partial_x - V\partial_y - U_x\end{pmatrix}\begin{pmatrix}\lambda^1\\\lambda^2\end{pmatrix}d\mathbf{x}$$

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

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

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:

$$\frac{\partial \mathbb{H}}{\partial \mathbf{W}} = \left(\frac{\partial \mathbb{H}}{\partial U} \ \frac{\partial \mathbb{H}}{\partial V}\right)$$

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 $\dfrac{\partial \mathbb{H}}{\partial V}\eta^2 = I_y\eta^2$ 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 (44) has two components and can be written as:

$$\frac{\partial U}{\partial t} + UU_x + VU_y = 0 \tag{55}$$

$$\frac{\partial V}{\partial t} + UV_x + VV_y = 0 \tag{56}$$

Equation (55) is rewritten and split into the system:

$$F = \frac{1}{2}U^2 \tag{57}$$

$$\frac{\partial U}{\partial t} + F_x = 0 \tag{58}$$

$$\frac{\partial U}{\partial t} + VU_y = 0 \tag{59}$$

Equations (58) and (59) are now linear advection equations. They can be approximated using the following schemes:

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

$$\frac{U_{i,j}^{k+1} - U_{i,j}^k}{\triangle t} = -S_y(V,U)_{i,j}^k$$

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 (56) is rewritten and split into the system:

$$G = \frac{1}{2}V^2$$

$$\frac{\partial V}{\partial t} + G_y = 0$$

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

and then approximated by:

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

$$\frac{V_{i,j}^{k+1} - V_{i,j}^k}{\triangle t} = -S_x(U,V)_{i,j}^k$$

C.2 Evolution equation of the adjoint variable

Equation (45) has two components and can be rewritten as follow:

$$-\lambda_t^1 - U\lambda_x^1 - V_y\lambda^1 - V\lambda_y^1 + V_x\lambda^2 = I_x A \tag{60}$$

$$-\lambda_t^2 - U\lambda_x^2 - U_x\lambda^2 - V\lambda_y^2 + U_y\lambda^1 = I_y A \tag{61}$$

with $A = -R^{-1} \star L$. Equation (60) is split into:

$$-\lambda_t^1 = U\lambda_x^1 + V_y\lambda^1$$

$$-\lambda_t^1 = V\lambda_y^1 - V_x\lambda^2 + I_x A$$

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:

$$-\frac{(\lambda^1)_{i,j}^k - (\lambda^1)_{i,j}^{k-1}}{\triangle t} = S_x(U,\lambda^1)_{i,j}^k + \frac{1}{2}(V_{i,j+1}^k - V_{i,j-1}^k)(\lambda^1)_{i,j}^k$$

$$-\frac{(\lambda^1)_{i,j}^k - (\lambda^1)_{i,j}^{k-1}}{\triangle t} = S_y(V,\lambda^1)_{i,j}^k - \frac{1}{2}(V_{i+1,j}^k - V_{i-1,j}^k)(\lambda^2)_{i,j}^k + (I_x A)_{i,j}^k$$

The numerical scheme is written:

$$(\lambda^1)^{k-1}_{i,j} = (\lambda^1)^k_{i,j} + \triangle t \left(S_x(U, \lambda^1)^k_{i,j} + \frac{1}{2}(V^k_{i,j+1} - V^k_{i,j-1})(\lambda^1)^k_{i,j} \right)$$

$$(\lambda^1)^{k-1}_{i,j} = (\lambda^1)^k_{i,j} + \triangle t \left(S_y(V, \lambda^1)^k_{i,j} - \frac{1}{2}(V^k_{i+1,j} - V^k_{i-1,j})(\lambda^2)^k_{i,j} + (I_x A)^k_{i,j} \right)$$

The equation (61) is split into:

$$-\lambda^2_t = U\lambda^2_x + U_x\lambda^2$$
$$-\lambda^2_t = V\lambda^2_y - U_y\lambda^1 + (I_y A)$$

and the numerical scheme is:

$$(\lambda^2)^{k-1}_{i,j} = (\lambda^2)^k_{i,j} + \triangle t \left(S_x(U, \lambda^2)^k_{i,j} + \frac{1}{2}(U^k_{i+1,j} - U^k_{i-1,j})(\lambda^2)^k_{i,j} \right)$$

$$(\lambda^2)^{k-1}_{i,j} = (\lambda^2)^k_{i,j} + \triangle t \left(S_y(V, \lambda^2)^k_{i,j} - \frac{1}{2}(U^k_{i,j+1} - U^k_{i,j-1})(\lambda^1)^k_{i,j} + (I_y A)^k_{i,j} \right)$$

C.3 Evolution equation of the incremental variable

Equation (46) has two components which are expressed as follow:

$$\delta U_t + U\delta U_x + V\delta U_y + U_x\delta U + U_y\delta V = Q \star \lambda^1 \qquad (62)$$
$$\delta V_t + U\delta V_x + V\delta V_y + V_x\delta U + V_y\delta V = Q \star \lambda^2 \qquad (63)$$

Equation (62) is split into:

$$\delta U_t + U\delta U_x + U_x\delta U = 0$$
$$\delta U_t + V\delta U_y + U_y\delta V = Q \star \lambda^1$$

Again, linear advection terms are approximated using shock filter.

$$\frac{\delta U^{k+1}_{i,j} - \delta U^k_{i,j}}{\triangle t} = -S_x(U, \delta U)^k_{i,j} - \frac{1}{2}(U^k_{i+1,j} - U^k_{i-1,j})\delta U^k_{i,j}$$

$$\frac{\delta U^{k+1}_{i,j} - \delta U^k_{i,j}}{\triangle t} = -S_y(V, \delta U)^k_{i,j} - \frac{1}{2}(U^k_{i,j+1} - U^k_{i,j-1})\delta V^k_{i,j} + (Q \star \lambda^1)^k_{i,j}$$

Equation (63) is split into:

$$\delta V_t + U\delta V_x + V_y\delta V = 0$$
$$\delta V_t + V\delta V_y + V_x\delta U = Q \star \lambda^2$$

and approximated by:

$$\frac{\delta V^{k+1}_{i,j} - \delta V^k_{i,j}}{\triangle t} = -S_x(U, \delta V)^k_{i,j} - \frac{1}{2}(V^k_{i,j+1} - V^k_{i,j-1})\delta V^k_{i,j}$$

$$\frac{\delta V^{k+1}_{i,j} - \delta V^k_{i,j}}{\triangle t} = -S_y(V, \delta V)^k_{i,j} - \frac{1}{2}(V^k_{i+1,j} - V^k_{i-1,j})\delta U^k_{i,j} + (Q \star \lambda^2)^k_{i,j}$$

References

1. L. Alvarez, J. Weickert, and J. Sánchez. Reliable estimation of dense optical flow fields with large displacements. International Journal of Computer Vision, 39(1):41–56, 2000.
2. A. Apte, C.K.R.T. Jones, A.M. Stuart, and J. Voss. Data assimilation: Mathematical and statistical perspectives. Int. J. Numer. Meth. Fluids, 56:1033–1046, 2008.
3. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. 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, May 2004.
4. J. Hadamard. Lecture on Cauchy's Problem in Linear Partial Differential Equations. Yale University Press, New Haven, 1923.
5. I. Herlin, F.-X. Le Dimet, E. Huot, and J.-P. Berroir. Coupling models and data: which possibilities for remotely-sensed images? In Poulicos Prastacos, Ulises Cortés, Juan-Luis Díaz De León, and Manuel Murillo, editors, e-Environment: Progress and Challenge, volume 11 of Research on Computing Science, pages 365–383. Instituto Politécnico Nacional, November 2004.
6. B.K.P. Horn and B.G. Schunk. Determining optical flow. Artificial Intelligence, 17:185–203, 1981.
7. E. Huot, I. Herlin, and G. Korotaev. 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., July 2008.
8. F. Le-Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects., pages 97–110. Tellus, 1986.
9. F.-X. Le Dimet, I.M. Navon, and D.N. Daescu. Second-order information in data assimilation. Monthly Weather Rev, 130:629–648, March 2002.
10. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics, XLII(577–685), 1989.
11. H.-H. Nagel. Displacement vectors derived from second-order intensity variations in image sequences. Computer Vision, Graphics, and Image Processing, 21:85–117, 1983.
12. J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. Signal Processing, 66(2):143–155, 1998.
13. D.S. Oliver. Calculation of the inverse of the covariance. Mathematical Geology, 30(7):911–933, 1998.
14. N. Papadakis, T. Corpetti, and É. Mémin. Dynamically consistent optical flow estimation. In Proceedings of International Conference on Computer Vision, Rio de Janeiro, Brazil, October 2007.
15. N. Papadakis, P. Héas, and É. Mémin. 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, November 2007.
16. N. Papadakis and É. Mémin. Variational optimal control technique for the tracking of deformable objects. In Proceedings of International Conference on Computer Vision, Rio de Janeiro, Brazil, October 2007.
17. N. Papadakis, E. Mémin, and F. Cao. A variational approach for object contour tracking. In Proceedings of ICCV'05 Workshop on Variational, Geometric and Level Set Methods in Computer Vision, Beijing, China, October 2005.
18. P. Perona and J. Malik. Space scale and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7):629–639, 1990.
19. M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In Proceedings of European Conference on Computer Vision, volume 2, pages 295–304, 1994.
20. J.A. Sethian. Level Set Methods. Cambridge University Press, 1996.
21. A. Tarantola. Inverse Problem Theory and Methods for Model Parameter Estimation. Society for Industrial and Applied Mathematics, 2005.
22. A. N. Tikhonov. Regularization of incorrectly posed problems. Sov. Math. Dokl., 4:1624–1627, 1963.
23. J.G. Verwer and B. Sportisse. A note on operator splitting in a stiff linear case. Technical Report MAS-R9830, Center voor Wiskunde en Informatica, December 1998.
24. J. Weickert. Anisotropic diffusion in image processing. ECMI Series. Teubner-Verlag, Stuttgart, 1998. ISBN:3-519-02606-6.

25. J. Weickert. Applications of nonlinear diffusion in image processing and computer vision. In Acta Math. Univ. Comenianae. Proceeding of Algoritmy 2000, volume LXX, pages 33–50, 2001.
26. J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. Journal of Mathematical Imaging and Vision, 14:245–255, 2001.
27. A. P. Witkin. Scale-space filtering. In Proc. 8th Int. Joint Conf. Art. Intell., pages 1019–1022, Karlsruhe, Germany, August 1983.