



HAL
open science

Assimilation de données pour estimer le mouvement et suivre un objet

Yann Lepoittevin, Isabelle Herlin, Dominique Béréziat

► **To cite this version:**

Yann Lepoittevin, Isabelle Herlin, Dominique Béréziat. Assimilation de données pour estimer le mouvement et suivre un objet. ORASIS - Journées francophones des jeunes chercheurs en vision par ordinateur, Le2i UMR CNRS 6306, Jun 2013, Cluny, France. hal-00826841

HAL Id: hal-00826841

<https://inria.hal.science/hal-00826841>

Submitted on 28 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Assimilation de données pour estimer le mouvement et suivre un objet

Data assimilation for motion estimation and object tracking

Yann Lepoittevin^{1,2*}

Isabelle Herlin^{1,2}

Dominique Béréziat³

¹ Inria

² CEREAs, Laboratoire commun École des Ponts ParisTech - EDF R&D, Université Paris-Est

³ Université Pierre et Marie Curie,

Inria, B.P. 105, 78153 Le Chesnay, France

yann.lepoittevin@inria.fr

Résumé

Cet article s'intéresse au problème de l'estimation du mouvement apparent sur une séquence d'images et au suivi d'un objet particulier. Pour ce faire, l'approche considérée est celle de l'assimilation de données, et, dans ce cas, de l'assimilation d'images. Cette approche repose sur les équations de la dynamique du système visualisé par la séquence d'images. Le modèle de dynamique considéré est la conservation lagrangienne de la vitesse et le transport des images et de la carte de distance qui modélise l'objet étudié. La méthode d'assimilation de données choisie, appelée 4D-Var, effectue une optimisation itérative d'une fonction de coût. Cette approche d'assimilation d'images est tout d'abord quantifiée au moyen d'une expérience jumelle, afin de démontrer l'amélioration obtenue sur l'estimation du mouvement. Puis la méthode est appliquée à une séquence d'images satellite météorologiques, avec l'objectif de suivre un nuage tropical. Les visualisations proposées illustrent les différentes composantes de la méthode.

Mots Clef

Assimilation de données, image satellite, mouvement, suivi d'objet.

Abstract

This paper studies the issues of computing apparent motion on an image sequence and tracking a given object. The mathematical approach is data assimilation, or in that context image assimilation, that relies on a dynamic model of the observed system. The chosen dynamic equations are the Lagrangian constancy of motion on pixels' trajectory and the transport of the image brightness function and of the distance map characterizing the tracked object. The data assimilation technic, named 4D-Var, is based on the iterative minimization of a cost function. This image assimilation method is first applied on synthetic data in order

to quantify results and measure the improvement on motion estimation. Then, the method is used on meteorological satellite acquisitions, with the objective to track a tropical cloud. Figures illustrate the main components of the approach.

Keywords

Data assimilation, satellite image, motion, object's tracking.

1 Introduction

Le suivi d'un objet dans une séquence d'images est un problème largement étudié dans la communauté scientifique de la vision par ordinateur. Une description détaillée de la littérature est disponible dans Yilmaz et al. [1], par exemple. Si les images étudiées sont bruitées, comme c'est le cas pour les acquisitions satellite, il est préférable d'introduire des hypothèses sur la dynamique du système observé afin d'obtenir un résultat de qualité. À notre connaissance, il n'existe actuellement pas de méthode permettant simultanément d'estimer le mouvement sur tout le domaine image et de suivre un objet particulier au cours de la séquence. Cependant, il existe des approches [2, 3, 4] qui segmentent et suivent un objet avec en entrée le champ de vitesse au cours de la séquence et la segmentation de l'objet sur la première image. D'autres [5] permettent de suivre une structure et d'estimer son déplacement à partir de sa segmentation initiale.

Les méthodes d'assimilation de données sont utilisées depuis une dizaine d'années dans la communauté du traitement d'image. Béréziat et al. [6] discutent le problème de l'estimation du mouvement et de l'utilisation des matrices de covariance pour filtrer les données bruitées ou manquantes. Papadakis et al. [7] décrivent une méthode d'assimilation de données variationnelle incrémentale, qui permet de calculer le mouvement et de suivre un objet en utilisant, comme données d'entrée, la séquence d'images

* Cette recherche est partiellement financée par la DGA

et la segmentation de l'objet sur l'intervalle temporel étudié. La méthode présentée dans cet article est plus complète et a pour objectif d'estimer le mouvement sur tout le domaine spatio-temporel, et de segmenter et suivre l'objet considéré. Les données d'entrée nécessaires à l'algorithme sont des données dites de bas niveau : la séquence d'images et les points de contour détectés sur ces images.

La section 2 permet de décrire les principales composantes mathématiques de la méthode : la modélisation de l'objet par une carte de distance, les lois d'évolution utilisées dans le modèle dynamique ainsi que la méthode d'assimilation de données, dite 4D-Var. La section 3 s'intéresse particulièrement à la technique de réinitialisation, utilisée pour permettre que l'advection d'une carte de distance reste une carte de distance lors d'une simulation numérique. La section 4 présente et analyse les résultats obtenus sur des séquences synthétiques, qui peuvent être quantifiés, et sur des images satellite météorologiques. Enfin, la section 5 permet d'analyser le travail réalisé et de proposer des perspectives pour ce travail de recherche.

2 Modélisation mathématique

Soit Ω un ouvert borné de \mathbb{R}^2 , dont la frontière est notée $\partial\Omega$. Soit $[t_0, T]$ un sous-ensemble borné de \mathbb{R} , qui correspond à l'intervalle temporel sur lequel des images sont disponibles. $\Omega_T := \Omega \times [t_0, T]$ est le domaine spatio-temporel étudié. $\langle \cdot, \cdot \rangle$ représente le produit scalaire associé à la norme $\|\cdot\|_H^2$ dans un espace de Hilbert H . $L_2(\Omega)$ est l'espace de fonctions $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ telles que :

$$\|f\|_{L_2}^2 := \int_{\Omega} f^2(x, y) dx dy < +\infty$$

$H^m(\Omega)$ est l'espace de Sobolev des fonctions $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ avec la norme :

$$\|f\|_{H^m}^2 := \sum_{\alpha_1 + \alpha_2 \leq m} \|\partial_x^{\alpha_1} \partial_y^{\alpha_2} f\|_{L_2}^2$$

sachant que $\partial_x = \frac{\partial}{\partial x}$ et $\partial_y = \frac{\partial}{\partial y}$ sont les dérivées partielles par rapport à x et y . $\mathbb{L}_2(t_0, T, H)$ est l'espace des fonctions $f : [t_0, T] \rightarrow H$, telles que :

$$\|f\|^2 := \int_{t_0}^T \|f(t)\|_H^2 dt < +\infty$$

Les images, les cartes de distance, et les deux composantes u et v du champ de mouvement \mathbf{w} sont définies sur $\mathbb{L}_2(t_0, T, H^1(\Omega))$. Un pixel de Ω est noté $\mathbf{x} = (x \ y)^T$ et le mouvement s'écrit : $\mathbf{w} = (u \ v)^T$. Dans cet article, nous notons u la fonction de $\mathbb{L}_2(t_0, T, H^1(\Omega))$, $u(t)$ la fonction de $H^1(\Omega)$ et $u(\mathbf{x}, t)$ sa valeur en un point \mathbf{x} de Ω . La même convention vaut pour toutes les grandeurs utilisées dans l'article.

2.1 Modélisation de l'objet

Un des objectifs de l'étude est de suivre un objet, par exemple un nuage, sur une séquence d'images satellite. Il

est difficile de déterminer et quantifier des propriétés de texture sur ce type d'objet. De même il n'est pas possible de définir des points caractéristiques dans le but de les suivre au cours du temps. Il faut de plus prendre en compte le fait que l'objet puisse se scinder en plusieurs composantes connexes sur la fenêtre temporelle étudiée. Le nuage est donc associé à sa courbe frontière $C(t)$, dont la topologie peut varier au cours du temps. Le modèle mathématique choisi pour représenter l'objet est une fonction implicite $\phi(t)$, dont les valeurs sont nulles sur la courbe $C(t)$ et représentent la distance signée à la courbe en tout autre point. Les valeurs sont donc positives à l'intérieur de l'objet et négatives à l'extérieur, comme illustré sur la figure 1. Cette fonction implicite $\phi(t)$ est appelée "carte de distance" dans la suite de l'article.

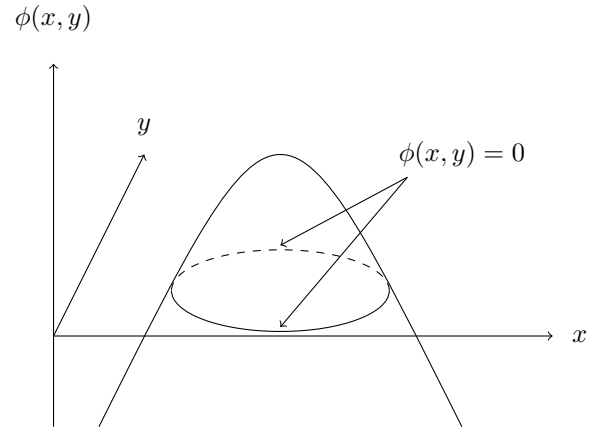


FIGURE 1 – Fonction implicite utilisée pour représenter la frontière de l'objet.

Les données d'entrée, qui sont les observations utilisées par la méthode d'assimilation, sont les images et les points de contour, calculés sur celles-ci au moyen d'un seuillage appliqué sur les maxima locaux de la norme du gradient [8].

2.2 Equations d'évolution

Des heuristiques sont utilisées pour définir le modèle dynamique sur lequel repose l'assimilation de données. Ces heuristiques concernent le champ de mouvement \mathbf{w} , les images et la carte de distance signée ϕ .

L'évolution du mouvement est supposée satisfaire l'hypothèse de constance lagrangienne, qui s'écrit :

$$\frac{du}{dt} = 0 \Leftrightarrow \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = 0 \quad (1)$$

$$\frac{dv}{dt} = 0 \Leftrightarrow \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = 0 \quad (2)$$

La méthode utilise une pseudo-image, notée I_s , qui a des propriétés physiques similaires aux acquisitions image et qui, en particulier, vérifie l'équation du flot optique :

$$\frac{\partial I_s}{\partial t} + \nabla I_s \cdot \mathbf{w} = 0 \quad (3)$$

où ∇ représente l'opérateur de gradient spatial. D'autres équations, telle que l'équation de conservation de la masse, ont été testées, sans avoir d'impact sur le résultat pour les images utilisées dans l'étude. L'intérêt de cette pseudo-image est qu'elle permet une comparaison aisée du modèle avec les acquisitions image au cours de l'assimilation : l'optimisation doit permettre d'obtenir que pseudo-images et acquisitions image deviennent presque identiques à la fin du processus. La fonction implicite ϕ est supposée satisfaire les mêmes heuristiques que I_s car l'objet se déplace de la même façon que les images :

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \mathbf{w} = 0 \quad (4)$$

Il est à présent possible de définir le vecteur d'état $\mathbf{X} = (u \ v \ I_s \ \phi)^T$, qui satisfait le modèle dynamique (1, 2, 3, 4), ou encore l'équation d'évolution :

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}(t)) = 0 \quad (5)$$

2.3 Assimilation de données

Estimer le mouvement \mathbf{w} sur la séquence et réaliser le suivi d'un objet caractérisé par une fonction implicite ϕ peut se faire en estimant le vecteur d'état \mathbf{X} au moyen d'un algorithme d'assimilation variationnelle, dit 4D-Var, qui repose sur les trois équations suivantes :

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{x}, t) + \mathbb{M}(\mathbf{X})(\mathbf{x}, t) = 0 \quad (6)$$

$$\mathbf{X}(\mathbf{x}, t_0) = \mathbf{X}_B(\mathbf{x}) + \varepsilon_B(\mathbf{x}) \quad (7)$$

$$\mathbb{H}(\mathbf{X}, \mathbf{Y})(\mathbf{x}, t) = \varepsilon_R(\mathbf{x}, t) \quad (8)$$

L'équation (6) s'appelle équation d'évolution, et son intégration temporelle permet d'obtenir la valeur de $\mathbf{X}(\mathbf{x}, t)$, pour tout t , à partir d'une condition initiale $\mathbf{X}(\mathbf{x}, 0)$.

L'équation (7), ou équation de background, permet d'exprimer la connaissance disponible, a priori, sur la valeur approximative $\mathbf{X}_B(\mathbf{x})$ que doit avoir le vecteur d'état $\mathbf{X}(\mathbf{x}, t_0)$ à la date t_0 . Le fait que ce ne soit qu'une connaissance approximative et que le résultat $\mathbf{X}(\mathbf{x}, t_0)$, estimé par assimilation de données, doive seulement rester proche de $\mathbf{X}_B(\mathbf{x})$, s'exprime par l'introduction d'un terme d'erreur $\varepsilon_B(\mathbf{x})$ dans cette équation. Etant donné qu'aucune information n'est disponible sur le champ de mouvement initial, l'objectif étant de l'estimer, le background du champ de mouvement est choisi avec une valeur nulle. Le background de la pseudo-image $I_s(t_0)$ est choisi égal à la première image de la séquence ; celui de $\phi(t_0)$, noté ϕ_B , est obtenu en calculant une carte de distance signée à partir d'une segmentation grossière de l'objet étudié, fournie par l'utilisateur. Notons \mathbb{P} la projection du vecteur d'état \mathbf{X} sur les deux composantes I_s et ϕ , l'équation (7) peut se réécrire de la façon suivante :

$$\mathbb{P}(\mathbf{X}(\mathbf{x}, t_0)) = \mathbb{P}(\mathbf{X}_B(\mathbf{x})) + \varepsilon_B(\mathbf{x}) \quad (9)$$

L'équation d'observation, exprimée par l'équation (8), traduit les liens entre les observations utilisées et le vecteur d'état \mathbf{X} . Dans le cas de notre étude, le vecteur d'observation \mathbf{Y} contient bien entendu les acquisitions image $I(t)$. Pour chacune d'elle, on calcule les points de contours, puis une carte de distance non signée, notée $D_c(t)$. En chaque pixel \mathbf{x} , $D_c(\mathbf{x}, t)$ représente la distance au point de contour le plus proche. Ces cartes de distance viennent compléter le vecteur d'observation qui s'écrit $\mathbf{Y} = (I \ D_c)^T$. Un opérateur d'observation \mathbb{H} est défini afin de comparer le vecteur d'état et les observations. \mathbb{H} s'écrit en deux parties : $\mathbb{H} = (\mathbb{H}_I \ \mathbb{H}_\phi)^T$. \mathbb{H}_I compare les pseudo-images I_s aux acquisitions images I :

$$\mathbb{H}_I(\mathbf{X}, \mathbf{Y})(\mathbf{x}, t) = I_s(\mathbf{x}, t) - I(\mathbf{x}, t) = \varepsilon_I(\mathbf{x}, t) \quad (10)$$

A la fin du processus d'assimilation, elles doivent être quasiment identiques, et leur différence est décrite par le terme d'erreur ε_I , qui doit donc être minimisé. \mathbb{H}_ϕ compare $\phi(\mathbf{x}, t)$ à la distance $D_c(\mathbf{x}, t)$. Comme ϕ est signée, sa valeur absolue doit être quasiment égale à D_c , afin que la frontière $C(t)$ de l'objet soit bien positionnée sur les points de contour. La courbe $C(t)$ est supposée ne pas se déplacer de plus de s pixels, d'un pas de temps au suivant, et est donc supposée rester à l'intérieur d'une région $\Omega_s(t)$. Il suffit alors de comparer D_c et $|\phi|$ dans le domaine $\Omega_s(t)$. On introduit la fonction de pondération $\zeta_s(\mathbf{x})$: $\zeta_s(\mathbf{x}) = 1$ si \mathbf{x} est situé sur la courbe $C(t)$, puis la valeur de la fonction décroît au fur et à mesure que \mathbf{x} s'éloigne de $C(t)$ et vérifie $\zeta_s(\mathbf{x}) < 10^{-5}$ en dehors de $\Omega_s(t)$. L'opérateur d'observation \mathbb{H}_ϕ s'écrit :

$$\begin{aligned} \mathbb{H}_\phi(\mathbf{X}, \mathbf{Y})(\mathbf{x}, t) &= \zeta_s(\mathbf{x}) (|\phi(\mathbf{x}, t)| - D_c(\mathbf{x}, t)) \\ &= \varepsilon_\phi(\mathbf{x}, t) \end{aligned} \quad (11)$$

Les termes d'erreurs ε_B , ε_I et ε_ϕ sont des variables gaussiennes, de moyenne nulle et non corrélés. La matrice de covariance de ε_B s'écrit B , et R_I et R_ϕ sont les variances respectives de ε_I et ε_ϕ . Pour résoudre le système (6, 9, 10, 11), il faut minimiser les erreurs ε_B , ε_I et ε_ϕ , et le problème se réécrit sous la forme de la minimisation d'une fonction de coût :

$$\begin{aligned} J(\mathbf{X}(t_0)) &= \int_{\Omega_T} \frac{\varepsilon_I(\mathbf{x}, t)^2}{R_I(\mathbf{x}, t)} d\mathbf{x}dt + \int_{\Omega_T} \frac{\varepsilon_\phi(\mathbf{x}, t)^2}{R_\phi(\mathbf{x}, t)} d\mathbf{x}dt \\ &\quad + \int_{\Omega} \varepsilon_B(\mathbf{x})^T B(\mathbf{x})^{-1} \varepsilon_B(\mathbf{x}) d\mathbf{x} \end{aligned} \quad (12)$$

Afin de mettre en place une méthode d'assimilation de données de type 4D-Var, une variable adjointe λ est définie par l'équation d'évolution (14), avec une valeur initiale nulle à la fin de la fenêtre temporelle étudiée (équation (13)) :

$$\lambda(T) = 0 \quad (13)$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}} \right)^* \lambda = \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}} \right)^* R^{-1} \mathbb{H}(\mathbf{X}, \mathbf{Y}) \quad (14)$$

$(\frac{\partial \mathbb{M}}{\partial \mathbf{X}})^*$ et $(\frac{\partial \mathbb{H}}{\partial \mathbf{X}})^*$ sont les opérateurs adjoints de $\frac{\partial \mathbb{M}}{\partial \mathbf{X}}$ et $\frac{\partial \mathbb{H}}{\partial \mathbf{X}}$ et vérifient : $\langle Z\eta, \mu \rangle = \langle \eta, Z^*\mu \rangle$ pour tout couple de fonctions (η, μ) . L'opérateur adjoint $(\frac{\partial \mathbb{M}}{\partial \mathbf{X}})^*$ est obtenu automatiquement, à partir de l'opérateur discret \mathbb{M} , au moyen du logiciel de différentiation automatique Tapenade [9]. Le calcul de la dérivée de J par rapport à $\mathbf{X}(t_0)$, utilisée pour la minimisation par descente de gradient, fournit :

$$\frac{\partial J}{\partial \mathbf{X}(t_0)} = \mathbb{P}^T B^{-1} [\mathbb{P}(\mathbf{X}(t_0)) - \mathbb{P}(\mathbf{X}_B)] + \lambda(t_0) \quad (15)$$

La minimisation itérative de la fonction de coût est effectuée par l'algorithme L-BFGS [10].

3 Module de réinitialisation

L'étude porte sur le suivi d'un objet caractérisé, à chaque date t , par une courbe $C(t)$. Le background de cette courbe, pour le processus d'assimilation, est une courbe fermée C_B , définie par l'utilisateur sur la première image de la séquence en segmentant grossièrement l'objet. Une carte de distance $\phi(t)$, associée à $C(t)$, est advectée par le champ de mouvement $\mathbf{w}(t)$ selon l'équation (4). La plupart des schémas numériques d'advection perdent immédiatement la propriété de carte de distance, qui s'écrit : $\|\nabla\phi(\mathbf{x}, t)\| = 1$ pour tout point \mathbf{x} . Il faut donc définir une méthode qui assure que cette propriété reste vérifiée au cours de l'intégration temporelle de l'équation (4). De plus, étant donné que $C(t)$, définie par $C(t) = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\}$, est initialisée par la courbe fermée C_B , la méthode développée doit assurer que $C(t)$ reste une courbe fermée au cours de l'advection.

Deux approches ont été étudiées pour assurer ces propriétés. La première approche consiste à définir un champ de mouvement modifié $\mathbf{w}_m(t)$, calculé à partir du champ de vitesse $\mathbf{w}(t)$, et d'utiliser $\mathbf{w}_m(t)$ pour advecter $\phi(t)$ en préservant la propriété $\|\nabla\phi(\mathbf{x}, t)\| = 1$. Un champ \mathbf{w}_m , possédant cette propriété, peut être calculé sur tout le domaine image au moyen de la méthode dite de Fast Marching [11]. En utilisant de plus les schémas de discrétisation de Sethian [11], $C(t)$ reste une courbe fermée. La méthode de Fast Marching a donc été examinée comme candidate pour notre système d'assimilation d'images. Toutefois, la mise en place d'un algorithme d'assimilation variationnelle 4D-Var nécessite le calcul et l'utilisation de son opérateur adjoint, comme expliqué par Le Dimet et al. [12]. L'adjointisation du Fast Marching nécessite de conserver en mémoire un tableau avec l'ordre dans lequel les pixels de l'image sont examinés. Le processus d'assimilation effectue ensuite de nombreux appels à ce tableau au cours des calculs. La conséquence immédiate est un besoin important en capacité mémoire et calcul, et une impossibilité de parallélisation. Nous avons donc abandonné l'idée d'utiliser un champ modifié et avons considéré la seconde approche, basée sur un module de réinitialisation, tel que proposé par Sussman et al. [13] : à chaque date t , $\phi(t)$ est tout d'abord advecté par $\mathbf{w}(t)$. Le résultat obtenu n'étant plus une carte de distance, il est modifié par le module de réinitialisation

afin de retrouver la propriété $\|\nabla\phi(\mathbf{x}, t)\| = 1$.

Expliquons donc comment fonctionne ce module de réinitialisation. A une date t donnée, notons ϕ_t la valeur de la fonction $\phi(t)$ obtenue en advectant $\phi(t-1)$ par le champ de mouvement $\mathbf{w}(t-1)$. Cette valeur ϕ_t n'est pas une carte de distance, et le module de réinitialisation doit être utilisé pour rétablir cette propriété. Il s'agit pour cela de trouver la valeur limite de l'équation différentielle partielle :

$$\frac{\partial \psi}{\partial \tau} = \text{sign}(\phi_t)(1 - \|\nabla\psi\|) \quad (16)$$

avec la condition initiale : $\psi(0) = \phi_t$. Le résultat obtenu, noté ψ_l , est une carte de distance, puisqu'il vérifie $\|\nabla\psi_l(\mathbf{x})\| = 1$ pour tout \mathbf{x} . Si le processus de réinitialisation arrive de plus à forcer ψ_l à avoir la même isocourbe de valeur 0 que ϕ_t , alors ψ_l est une carte de distance qui caractérise la courbe $C(t)$. Il suffit donc de choisir ψ_l pour valeur de $\phi(t)$ avant de poursuivre l'intégration de t à $t+1$. La question de l'isocourbe de valeur 0 sera discutée dans le paragraphe suivant. Il faut rappeler que la fonction ϕ ne doit être calculée que dans la région $\Omega_s(t)$, qui entoure la courbe frontière de l'objet, puisqu'on considère que le déplacement de l'objet d'une image à la suivante est borné par la valeur s . Le processus de réinitialisation, réalisé par l'intégration de l'équation (16), s'arrête donc quand la propriété $\|\nabla\psi_l(\mathbf{x})\| = 1$ est vérifiée pour tout \mathbf{x} tel que $|\psi_l(\mathbf{x})| < s$. D'après Sussman et al. [13], il faut pour cela $\frac{s}{\Delta\tau}$ intégrations de l'équation (16), où $\Delta\tau$ représente le pas de discrétisation de la variable τ . Puisque le nombre de pas d'intégration est connu à l'avance, il devient possible d'écrire le programme de façon à ce que l'utilisation de l'adjoint par la méthode 4D-Var devienne raisonnable, en termes de capacité mémoire et temps de calcul.

Revenons à la courbe d'isovaleur nulle. Dans le cas continu, Sussman et al. [13] ont démontré que ψ_l a la même courbe d'isovaleur nulle que ϕ_t , appelée C_t . Toutefois, cette propriété n'est plus vérifiée après discrétisation. Définissons C_ϵ , un petit anneau de taille 2ϵ autour de C_t . Notons P_{ij} le pixel d'indices i et j . Au cours de l'intégration de l'équation (16), il faut que la valeur ψ_{ij} de ψ soit proche de la valeur $\phi_{t,ij}$, pour tout pixel P_{ij} qui intersecte l'anneau C_ϵ . De même, la valeur $\|\nabla\psi_{ij}\|$, dans C_ϵ , doit rester proche de $\|\nabla\phi_{t,ij}\|$. La solution est d'ajouter un terme de contrainte $\mu f(\phi_t)$ dans le terme de droite de l'équation (16), de façon à ce que $\text{sign}(\phi_t)(1 - \|\nabla\psi\|) + \mu f(\phi_t)$ soit de valeur quasi nulle dans C_ϵ . Ainsi la valeur de $\psi(x)$ n'évolue pas l'anneau C_ϵ . L'équation (16) devient :

$$\frac{\partial \psi}{\partial \tau} = \text{sign}(\phi_t)(1 - \|\nabla\psi\|) + \mu f(\phi_t) \quad (17)$$

Avant de définir μ et f , il faut caractériser mathématiquement l'anneau C_ϵ au moyen d'une approximation de l'in-

dicatrice, obtenue comme dérivée de la fonction H_ϵ :

$$H_\epsilon(\psi) = \begin{cases} 0 & \text{si } \psi(\mathbf{x}) < -\epsilon \\ \frac{1}{2}\left(1 + \frac{\psi}{\epsilon} - \frac{1}{\pi} \sin\left(\pi \frac{\psi}{\epsilon}\right)\right) & \text{si } |\psi(\mathbf{x})| < \epsilon \\ 1 & \text{si } \psi(\mathbf{x}) > \epsilon \end{cases} \quad (18)$$

Afin que les valeurs $\|\nabla\psi_{ij}\|$ restent également constantes et égales aux valeurs de $\|\nabla\phi_{t,ij}\|$, au cours de l'intégration, pour les pixels P_{ij} intersectant C_ϵ , il faut de manière équivalente garder constantes les valeurs $H'_\epsilon(\psi_{ij})\|\nabla\psi(x_{ij})\|$, sachant que ces valeurs discrètes sont obtenues en intégrant la fonction continue dans les pixels P_{ij} :

$$H'_\epsilon(\psi)\|\nabla\psi_{ij}\| = \frac{\int_{P_{ij}} H'_\epsilon(\psi)\|\nabla\psi(\mathbf{x})\|d\mathbf{x}}{\int_{P_{ij}} H'_\epsilon(\psi)d\mathbf{x}} \quad (19)$$

On définit : $f(\phi_t) = \|\nabla\phi_t\|$. La valeur théorique de μ doit alors être choisie pour que le terme droit de l'équation (17) soit quasi nul dans C_ϵ :

$$\mu = \frac{-\int_{P_{ij}} H'_\epsilon(\phi_t)(\text{sign}(\phi_t)(1 - \|\nabla\psi\|))d\mathbf{x}}{\int_{P_{ij}} H'_\epsilon(\phi_t)f(\phi_t)d\mathbf{x}} \quad (20)$$

Ce module de réinitialisation est ajouté à l'évolution de ϕ dans l'équation (4). Après discrétisation spatio-temporelle, l'intégration du modèle d'évolution (5) s'écrit :

1. Initialisation de $\mathbf{w}(0)$, $I_s(0)$ et $\phi(0)$.
2. Pour $t = 0, \dots, T - 1$
 - (a) Réaliser l'advection par $\mathbf{w}(t)$ de $\mathbf{w}(t)$, $I_s(t)$, $\phi(t)$ afin d'obtenir $\mathbf{w}(t + 1)$, $I_s(t + 1)$, ϕ_{t+1} .
 - (b) En prenant ϕ_{t+1} comme valeur initiale, réaliser $\frac{s}{\Delta\tau}$ pas d'intégration du processus de réinitialisation en utilisant la loi d'évolution de ψ décrite par l'équation (17).
 - (c) Prendre $\phi(t + 1) = \psi_l$.

4 Résultats

4.1 Expérience jumelle

Dans un premier temps, la méthode présentée dans l'article est testée sur une expérience synthétique. Une séquence de 18 observations $I_i = I(t_i)$ pour $i = 1$ à 18 est obtenue par intégration du modèle d'évolution \mathbb{M} de l'équation (5), à partir des conditions initiales, champ de vitesse synthétique et image satellite, visualisées sur la Figure 2. Quatre des 18 observations I_i sont visibles sur la Figure 3. Les contours, puis les cartes de distance $D_c(\mathbf{x}, t_i)$, sont calculés sur les images I_i afin d'obtenir le vecteur d'observation \mathbf{Y} . Une expérience d'assimilation, telle que décrite dans la sous-section 2.3, est réalisée afin d'estimer le champ de mouvement et de suivre le nuage. L'intérêt de ce type d'expérience, appelée expérience jumelle, est d'avoir une connaissance complète de l'état du système et en particulier des champs de mouvement aux différentes dates. Après

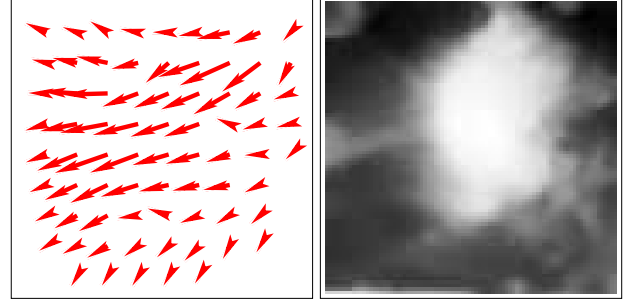


FIGURE 2 – Gauche : Champ de vitesse initial. Droite : Image initiale.

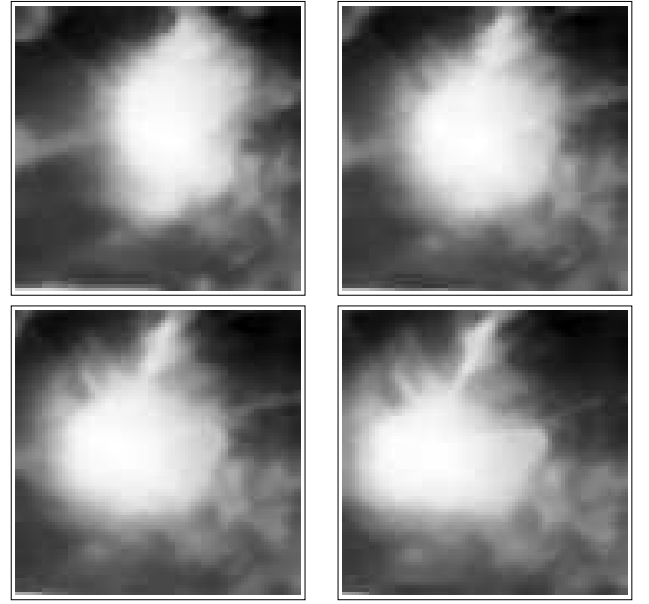


FIGURE 3 – De haut en bas, de gauche à droite : Observations I_2 , I_7 , I_{12} , I_{17} .

avoir réalisé l'assimilation de données, il est donc possible d'évaluer sa pertinence. On appelle \mathbf{w}_e le champ estimé et \mathbf{w}_r la vérité terrain. Des statistiques sur la différence entre \mathbf{w}_e et \mathbf{w}_r , sont fournies dans la table 1. La première colonne concerne le résultat obtenu par notre approche, tandis que la seconde présente l'erreur obtenue si le mouvement est estimé sans mettre une composante ϕ dans le vecteur d'état \mathbf{X} . Notre méthode est difficilement comparable

	avec ϕ	sans ϕ
Erreur sur la norme (en $m.s^{-1}$) : $ \ \mathbf{w}_e\ - \ \mathbf{w}_r\ $	0.1	0.3
Erreur angulaire (en degré)	9.6	13.0

TABLE 1 – Moyenne des erreurs entre champ estimé \mathbf{w}_e et vérité terrain \mathbf{w}_r . La valeur moyenne de la norme de \mathbf{w}_r est de $1.9 m.s^{-1}$.

à celles de l'état de l'art. Le modèle d'évolution tempo-

rel fait que nous utilisons un champ de vitesse différent, à chaque pas de discrétisation de la fenêtre temporelle, pour réaliser l'advection I_s et ϕ . D'un autre côté, les méthodes de flot optique donnent un champ unique entre deux dates d'observation. Afin de faire une analyse la plus objective possible, nous avons choisi d'advecter 8 points au moyen des différents champs de mouvement. Il est alors possible de suivre le déplacement de ces points, sur la fenêtre temporelle, et de comparer leurs positions à la date finale. La figure 4 montre, en rouge, la position des points pour la vérité terrain, en bleu les résultats obtenus avec notre méthode et en vert les résultats de la méthode de flot optique décrite par Sun et al. [14]. Une ellipse bleue caractérise un point pour lequel notre méthode permet une meilleure estimation du mouvement, une ellipse grise correspond au cas où notre méthode est équivalente à celle de Sun, et une ellipse verte au cas où le résultat de Sun est meilleur. Notre méthode fournit un meilleur résultat sur 5 points, elle est équivalente à Sun pour 2 points, et le flot optique est le meilleur pour 1 point. En moyenne, notre erreur de position à la date finale est de 2,7 pixels, alors que cette erreur moyenne est de 5 pixels pour le flot optique. Ce résultat se renouvelle en choisissant d'autres points sur l'image.

La figure 5 a pour objectif d'illustrer le processus de réinitialisation et montre les valeurs de $\|\nabla\psi(\mathbf{x})\|$ au cours des itérations de l'équation (17). L'image en haut à gauche fournit la norme du gradient de ϕ_1 , obtenu par advection de $\phi(0)$. On note que ce gradient n'est pas uniformément égal à 1, car la couleur n'est pas uniformément bleue. Les quatre images suivantes présentent plusieurs étapes du processus de réinitialisation, dans le domaine Ω_s . A la dernière itération, la propriété caractérisant une carte de distance est de nouveau valable : $\|\nabla\psi_l(\mathbf{x})\| = 1$, sauf pour les pixels \mathbf{x} de C_ϵ , qui ne sont pas évalués lors de la réinitialisation. On pose donc $\phi(1) = \psi_l$ avant de poursuivre l'intégration temporelle de l'équation (6). La figure 6 illustre également ce processus de réinitialisation pour 4 pixels choisis aléatoirement dans l'image. On présente en abscisse le nombre d'itérations du processus de réinitialisation et en ordonnée la norme du gradient. Pour cette visualisation, le pas de discrétisation du domaine a été diminué, afin d'augmenter le nombre d'itérations et de mieux comprendre le processus. On remarque que les 4 courbes convergent vers une valeur de norme égale à 1.

4.2 Images satellite

L'approche est ensuite appliquée à une séquence de 18 acquisitions satellite dont trois images sont visibles sur la colonne de gauche de la figure 7. Il s'agit des images I_2 , I_{10} et I_{18} . Cette figure visualise également les points de contour, en vert, sur la colonne centrale et le résultat de la méthode de suivi d'objet par une courbe rouge sur la colonne de droite. Cette courbe rouge représente donc les valeurs nulles de ϕ aux dates 2, 10 et 18. Afin de mieux

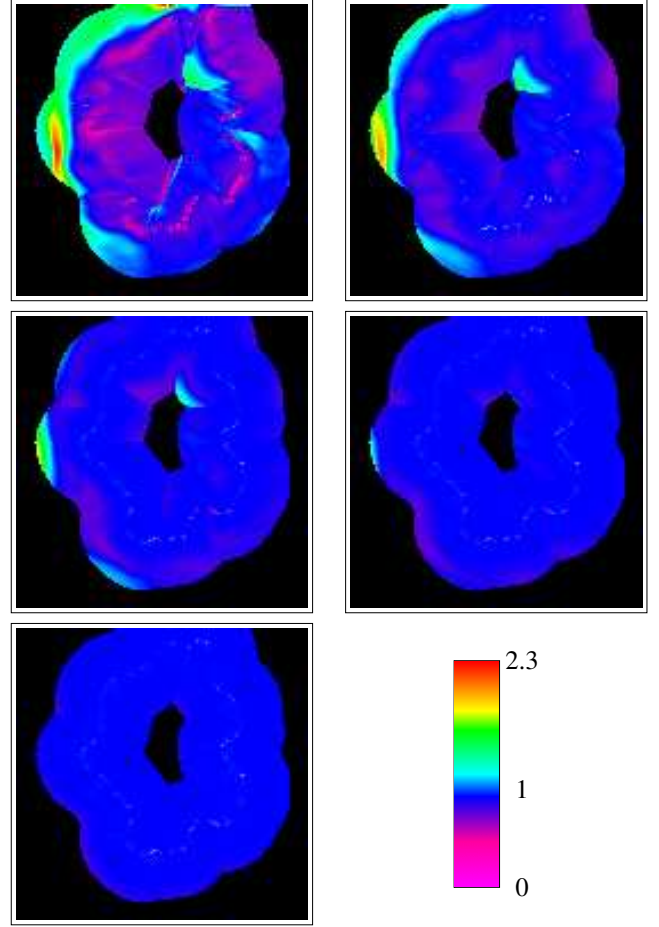


FIGURE 5 – De haut en bas et de gauche à droite : valeurs de la norme du gradient de $\psi(\mathbf{x})$ pour plusieurs itérations du processus de réinitialisation, dans le domaine Ω_s .

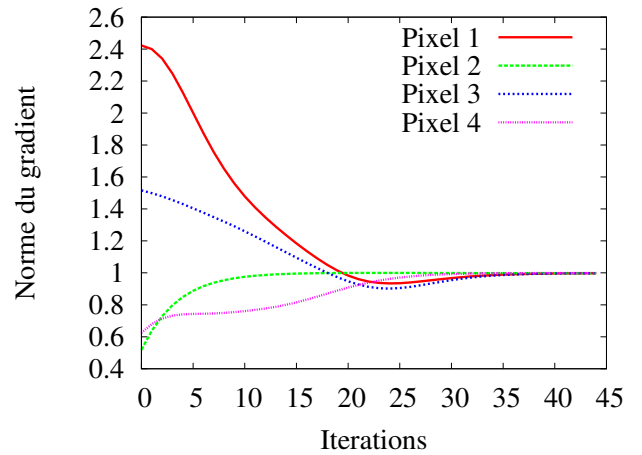


FIGURE 6 – Evolution de la norme du gradient, en 4 pixels, au cours des itérations de la réinitialisation.

comprendre l'intérêt d'inclure une composante ϕ dans le vecteur d'état \mathbf{Y} , nous avons également visualisé, en bleu, le résultat de l'advection de $\phi(0)$, segmentation obtenue

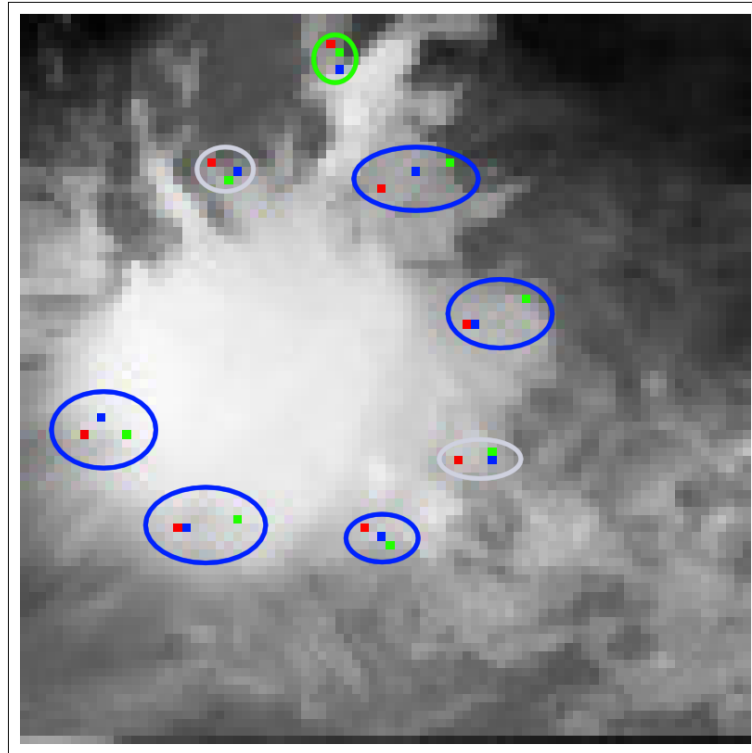


FIGURE 4 – Point rouge : vérité terrain. Point bleu : notre méthode. Point vert : flot optique de Sun et al. [14]. Ellipse bleue : notre méthode est la meilleure. Ellipse verte : le résultat de Sun est meilleur. Ellipse grise : résultats équivalents.

par notre approche à la date 0, par le champ de mouvement calculé sans prendre en compte ϕ dans le vecteur d'état. L'amélioration de l'estimation du mouvement grâce à la prise en compte de ϕ est clairement visible.

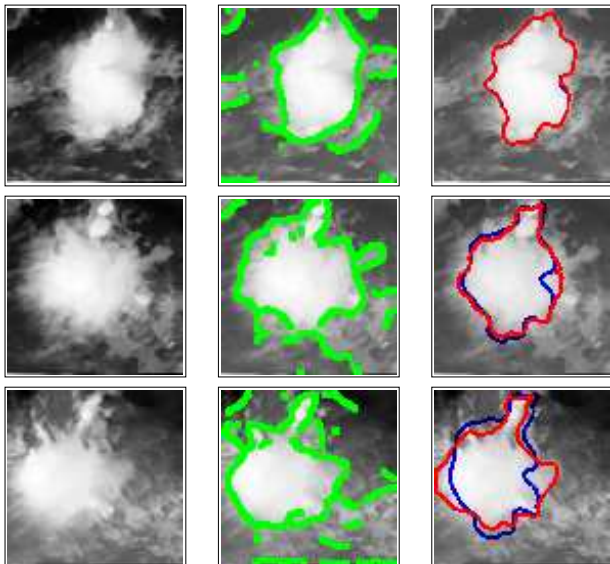


FIGURE 7 – Gauche : images satellite. Milieu : les points de contours en vert. Droite : la courbe rouge correspond au mouvement obtenu avec ϕ , et la bleue sans ϕ .

Afin d'illustrer l'intérêt du processus de réinitialisation, la

Figure 8 montre, en couleur, les valeurs d'une fonction ϕ pour $\phi(\mathbf{x}) \in [-6, 0]$. La valeur 0 est en rose ; -1 en bleu ; -2 en cyan ; -3 en vert ; -4 en jaune ; -6 en rouge. En haut, la fonction ϕ visualisée est une carte de distance. Les deux représentations du bas visualisent le résultat obtenu après 10 itérations du processus d'advection. A gauche, on voit le résultat obtenu si l'advection utilise le module de réinitialisation : ϕ reste une carte de distance. A droite, le résultat obtenu sans réinitialisation n'est plus une carte de distance : les valeurs de -3 à -6 ont quasiment disparues.

5 CONCLUSION

L'article décrit une approche originale qui permet, simultanément, d'estimer le mouvement sur une séquence d'images et de suivre un objet particulier sur celle-ci. L'application traitée dans l'article est le suivi d'un nuage tropical sur les images satellite météorologiques.

L'approche est basée sur une technique d'assimilation variationnelle de type 4D-Var. Le vecteur d'état inclut le champ de mouvement, les images et une fonction implicite ϕ , qui modélise la frontière de l'objet suivi. Le modèle d'évolution temporel choisi correspond à la conservation lagrangienne de la vitesse et au transport de l'image et de la carte de distance. Un module de réinitialisation a été développé afin que l'advection de la fonction implicite ϕ soit réalisée en conservant la propriété de carte de distance.

La méthode a été testée sur une expérience synthétique et sur des données météorologiques réelles. Dans les deux

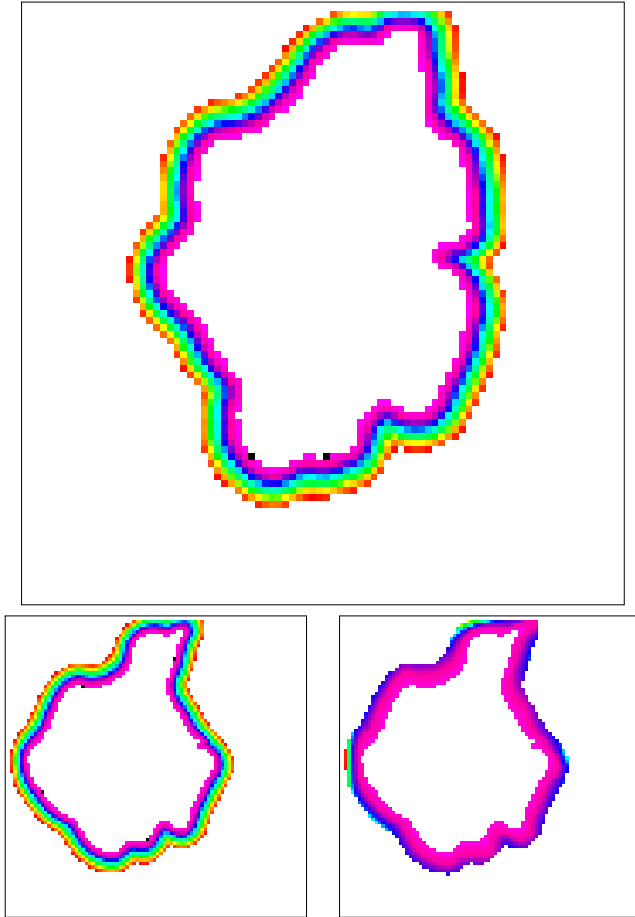


FIGURE 8 – Haut : Carte de distance ϕ visualisée pour $\phi(\mathbf{x}) \in [-6, 0]$. Bas : Résultat après 10 itérations. Gauche : avec le module de réinitialisation. Droite : sans réinitialisation.

cas, les résultats démontrent que le mouvement est mieux estimé si on inclut ϕ dans le vecteur d'état. L'utilisation du processus de réinitialisation pour recouvrir la propriété carte de distance a également été illustré. Malgré des temps de calcul longs, les résultats montrent que l'estimation du mouvement par cette méthode d'assimilation de données est plus précise que par un flot optique.

Les perspectives de recherche concernent principalement l'extension au suivi de plusieurs objets et à un processus de segmentation spatio-temporel. Un autre point d'étude envisagé est l'utilisation d'une technique d'assimilation variationnelle dite faible, qui utilise un terme d'erreur dans l'équation du modèle (6). Des méthodes d'assimilation d'ensemble sont envisagées afin de permettre une parallélisation et des temps de calcul réduits.

Références

- [1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking : A survey," *ACM Computing Surveys*, vol. 38, no. 4 2006, pp. 13, 2006. 1
- [2] N. Peterfreund, "Robust tracking of position and velocity with Kalman snakes," *IEEE Transactions on Pattern Ana-*

- lysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, June 1999. 1
- [3] Y. Rathi, N. Vaswani, A. Tannenbaum, and A.J. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, August 2007. 1
- [4] C. Avenel, É. Mémin, and P. Pérez, "Tracking closed curves with non-linear stochastic filters.," in *Conference on Space-Scale and Variational Methods*, 2009. 1
- [5] M. Bertalmío, G. Sapiro, and G. Randall, "Morphing active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 733–737, July 2000. 1
- [6] D. Béréziat and I. Herlin, "Solving ill-posed image processing problems using data assimilation," *Numerical Algorithms*, vol. 52, no. 2, pp. 219–252, 2011. 1
- [7] N. Papadakis and É. Mémin, "A variational technique for time consistent tracking of curves and motion," *Journal of Mathematical Imaging and Vision*, 2008. 1
- [8] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," *International Journal of Computer Vision*, vol. 1, no. 2, pp. 167–187, 1987. 2
- [9] L. Hascoët and V. Pascual, "Tapenade 2.1 user's guide," Technical Report 0300, INRIA, 2004. 4
- [10] C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal, "L-BFGS-B : Algorithm 778 : L-BFGS-B, FORTRAN routines for large scale bound constrained optimization," *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, 1997. 4
- [11] J.A. Sethian, *Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1999. 4
- [12] F. Le Dimet and O. Talagrand, *Variational algorithms for analysis and assimilation of meteorological observations : theoretical aspects.*, pp. 97–110, Tellus, 1986. 4
- [13] M. Sussman and E. Fatemi, "An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow," *Society for Industrial and Applied Mathematics*, 1999. 4
- [14] D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *Proceedings of European Conference on Computer Vision*, 2010, pp. 2432–2439. 6, 7