Estimation of motion from observed objects in image sequences

Yann Lepoittevin

Université Pierre et Marie Curie

École doctorale de Paris centre
Équipe projet Clime - Inria

Estimation de la dynamique à partir des structures observées dans une séquence d'images.

Par Yann Lepoittevin

Thèse de doctorat de Mathématiques appliquées
Dirigée par Isabelle Herlin et Dominique Béréziat

Présentée et soutenue publiquement le

Devant un jury composé de :

Pr Didier Auroux	Examinateur
Dr Dominique Béréziat	Co-directeur de thèse
Pr Fabrice Heitz	Examinateur
Dr Isabelle Herlin	Directeur de thèse
Dr Erwann Le Pennec	Examinateur
Pr Sylvie Thiria	Examinateur

Estimation de la dynamique à partir des structures observées dans une séquence d'images

Introduction

Les méthodes d'assimilation de données sont utilisées depuis une dizaine d'années dans la communauté du traitement d'image. Béréziat et al. [1] discutent ainsi le problème de l'estimation du mouvement par une méthode variationnelle incrémentale et l'utilisation des matrices de covariance d'erreur pour filtrer les données bruitées ou occultées. Les méthodes séquentielles d'assimilation, telles que le filtre de Kalman, utilisé par exemple dans la méthode décrite par Elad et al. [2], ont également été appliquées pour réaliser l'estimation du mouvement. Ce travail de thèse s'inscrit ainsi dans une double composante méthodologique, variationnelle et séquentielle, pour permettre l'estimation du mouvement à partir de séquences d'images.

Il est possible de prendre en compte les objets visibles dans les images lors de l'estimation du mouvement, aussi bien dans le cadre variationnel que séquentiel. Papadakis et al. [3] décrivent ainsi une méthode d'assimilation de données variationnelle, qui permet de calculer le mouvement en utilisant des informations sur les objets visualisés sur les données. Cependant, cette méthode nécessite la segmentation préalable des objets sur toutes les images de la séquence. Modéliser les objets et les inclure dans le processus d'estimation du mouvement, par une méthode d'assimilation de données, permet en fait de réaliser une corrélation en espace des informations caractérisant les pixels et donc de prendre en compte le déplacement des objets pour l'estimation du mouvement dans tout le domaine image. Les travaux effectués pendant cette thèse ont donc principalement porté sur la prise en compte des objets dans les processus d'assimilation d'images, sur la recherche d'une représentation optimale de ces objets, sur le choix d'une technique d'assimilation de données adaptée en fonction de l'application et enfin sur l'utilisation des travaux dans le contexte opérationnel de la prévision immédiate des pluies par imagerie radar.

Assimilation variationnelle

L'algorithme d'assimilation variationnelle, dit 4D-Var, repose sur trois équations : une équation d'évolution qui traduit la connaissance physique sur le phénomène étudié ; une équation d'ébauche, qui représente les connaissances a priori sur le système ; une équation d'observation, qui fait le lien entre le vecteur d'état, caractérisant le système étudié, et les observations, qui sont ici des acquisitions images et/ou des caractéristiques calculées sur ces acquisitions. Le problème à résoudre

s'écrit alors sous la forme de la minimisation d'une fonction de coût J, qui traduit la nécessité de minimiser simultanément les erreurs de l'équation d'évolution, de l'équation d'ébauche et de l'équation d'observation.

Dans le contexte de cette thèse, les principales composantes mathématiques des méthodes développées portent d'une part sur les différentes modélisations des objets visualisés sur la séquence d'images et, d'autre part, sur la définition des lois d'évolution utilisées dans le modèle dynamique.

La première partie de ce document de thèse, consacrée à l'assimilation variationnelle, a permis de mettre au point une méthode dont les données d'entrée sont des données dites de bas niveau : la séquence d'images et les points de contour détectés sur ces acquisitions. L'objet est associé à sa courbe frontière, dont la topologie peut varier au cours du temps (il peut s'agir en fait d'un ensemble d'objets). La représentation mathématique choisie est une fonction implicite, dont la valeur est nulle sur la courbe frontière et a pour valeur la distance signée à la courbe en tout autre point de l'image. Le modèle dynamique caractérise l'évolution des images et modifie la fonction implicite afin que la position de la courbe de niveau 0 corresponde à celle des points de contour détectés dans les acquisitions image. À chaque itération du minimiseur de la fonction de coût J, le modèle d'évolution temporelle est utilisé en intégration directe sur l'intervalle de temps étudié, ainsi que son adjoint en intégration rétrograde. La prise en compte des objets impacte donc fortement les temps de calcul et les contraintes de mémoire. C'est la principale faiblesse de cette méthode, qui permet toutefois une amélioration réelle de l'estimation du mouvement.

Lors du processus de minimisation de la fonction de coût, réalisé au cours de l'assimilation variationnelle, les valeurs des termes d'erreur d'évolution, d'ébauche et d'observation, pour des pixels différents, sont corrélées par l'utilisation de matrices de covariance d'erreur. Dans une première étape, ces matrices ont été définies comme étant diagonales, sans corrélation entre pixels. En effet, ces contraintes entre pixels différents sont prises en compte, dans le vecteur d'observation, par l'utilisation de la carte de distance aux points de contours. Dans une deuxième étape, la covariance entre pixels a été modélisée comme une fonction de la distance entre pixels et de leur proximité aux points de contour. Malheureusement, la taille de la matrice de covariance étant le carré de la taille de l'image multipliée par la taille du vecteur d'état, cette matrice ne peut être conservée en mémoire durant le processus de minimisation. La prise en compte des covariances nécessite donc une solution alternative. Nous avons alors choisi de réécrire la matrice de covariance d'erreur de l'ébauche du vecteur mouvement comme une fonction locale dépendant du pixel et de ses voisins. Cette matrice approxime le laplacien du mouvement et est ajoutée, sous la forme d'un terme de régularisation, à la fonction de coût J minimisée lors de l'assimilation.

Une partie du travail de thèse a été consacrée au développement d'un algorithme de prévision immédiate des précipitations, avec des échéances de moins d'une heure. Ce travail a été réalisé en collaboration avec l'entreprise Weather Measures [1], qui opère un ensemble de radars au sol, mesurant la réflectivité, sur la région de Clermont-Ferrand. La réflectivité donne une information sur les taux de précipitation et peut donc être utilisée pour faire une prévision des crues rapides, fréquentes dans la région, et en prévenir les impacts. Les algorithmes de l'état de l'art effectuant une prévision immmédiate se décomposent en deux phases : une phase d'estimation du mouvement des cellules de pluie, utilisant plusieurs acquisitions ; une phase de prévision, utilisant le résultat de l'estimation ainsi que la dernière acquisition, pour intégrer un modèle de simulation. Les algorithmes usuels diffèrent principalement sur les méthodes utilisées lors de la phase d'estimation. La plupart reposent sur des caractéristiques images et imposent de fortes contraintes, telle que l'identification

[1] `http://www.weather-measures.fr/`

des cellules de pluie d'une acquisition à la suivante. La méthode d'estimation proposée, modélisant les objets par une carte de distance signée, permet de s'affranchir des contraintes les plus fortes et d'obtenir une estimation dirigée par un suivi global des cellules de pluie et par l'évolution des images. La phase de prévision repose sur un modèle physique plus complet que celui utilisé pour l'estimation et permet d'obtenir des prévisions à l'échéance souhaitée. L'algorithme proposé, bien que nécessitant encore des améliorations, a permis, sur des situations réelles, d'obtenir des prévisions de bonne qualité.

Assimilation séquentielle

Le défaut principal des méthodes variationnelles est qu'elles sont coûteuses en temps de calcul. Des alternatives aux méthodes précédentes, basées sur l'assimilation de données séquentielle, ont donc été étudiées. Nous nous sommes ainsi intéressés au filtre de Kalman d'ensemble [4], qui nécessite la définition d'un modèle dynamique et la création d'un ensemble initial de vecteurs d'état modélisant les incertitudes a priori. L'utilisation du filtre de Kalman d'ensemble, pour définir la méthode appelée IEnKF, Image-base Ensemble Kalman Filter, nécessite également de concevoir des techniques de localisation spécifiques, rendant possible l'utilisation de la méthode sur des images réelles et potentiellement bruitées.

La méthode IEnKF présente l'intérêt de fournir une approximation de l'erreur d'estimation du champ de mouvement, en plus de l'estimation elle-même. Il faut toutefois noter qu'elle nécessite d'estimer l'incertitude sur l'état initial au moyen d'un ensemble de vecteurs d'état, qui caractérise les champs initiaux du mouvement et de l'image.

Dans l'approche que nous avons définie, la construction de l'ensemble initial repose sur le constat qu'il existe de nombreux algorithmes d'estimation de mouvement reposant sur différentes hypothèses, formulations et paramétrisations. Chacun d'entre eux est adapté à un contexte image différent. En considérant un ensemble d'algorithmes, il est alors possible de générer un ensemble de champs de mouvement entre les deux premières images de la séquence. L'ensemble de vecteurs d'état est alors obtenu en utilisant ces champs de mouvement et la première acquisition de la séquence d'images étudiée. Cet ensemble est intégré en temps par le modèle d'évolution jusqu'à la date de première observation. À cette date, l'ensemble des vecteurs d'état permet d'obtenir une approximation de la densité a priori, supposée gaussienne, par le calcul de la moyenne et de la matrice de covariance de l'ensemble. L'utilisation de l'observation permet de réaliser une analyse. Le champ de mouvement est estimé comme étant la moyenne de l'ensemble ainsi obtenu, et la répartition de cet ensemble fournit une approximation de l'erreur d'estimation.

L'application du filtre de Kalman d'ensemble sur des données réelles nécessite l'utilisation de techniques de localisation. En effet, l'approximation de la densité de probabilité du vecteur d'état, liée à l'échantillonnage de cette loi par les membres de l'ensemble, fait que l'ensemble se resserre, c'est-à-dire que la variance diminue, après prise en compte de l'observation lors de l'analyse. L'estimation du mouvement devient alors impossible aux dates ultérieures. Il est premièrement nécessaire d'utiliser des ensembles d'observations, échantillonnant l'erreur d'acquisition et de représentativité des observations utilisées pour l'assimilation de données, afin d'éviter que la matrice de covariance d'erreur sur l'état ne soit sous-évaluée après analyse. Il faut d'autre part recourir à des méthodes dites de localisation pour diminuer l'impact de grandes valeurs erronées dans la matrice de covariance d'erreur sur l'état, dues à l'échantillonnage de cette matrice. La localisation par critère de distance de la matrice de covariance est une technique usuelle, qui considère que des pixels

éloignés dans l'image doivent être moins corrélés que des pixels proches, que nous utilisons pour l'implémentation de IEnKF.

Un critère supplémentaire de localisation, portant sur l'intensité de niveau de gris des pixels, a été défini dans cette thèse, afin de permettre que des pixels proches, mais n'appartenant pas à une même structure, soient moins corrélés que des pixels plus éloignés, mais appartenant à la même structure.

Cependant, la localisation, si elle est effectuée directement sur la matrice de covariance, rend la méthode inutilisable pour des images de grande taille comme le sont les acquisitions satellite. Les matrices ne peuvent plus être stockées en mémoire à cause de leur taille et les temps de calcul nécessaires à leur inversion deviennent tout simplement prohibitifs. La localisation peut cependant s'effectuer par une méthode consistant à découper au préalable le domaine global en sous-domaines, sur lesquels l'analyse est calculée indépendamment des autres sous-domaines. La prise en compte des objets visibles sur les images intervient lors du découpage du domaine d'analyse global en sous-domaines.

Toutes les méthodes présentées dans ce document, aussi bien variationnelles que séquentielles, ont été testées sur des images synthétiques et sur des données réelles, images satellite ou de trafic urbain.

Contents

Introduction

The research work described in this document has been realized since mid 2012 as part of my PhD at Inria. Inria is the french National Institute for computer science and applied mathematics. Research at Inria is organized in "project teams", which bring together researchers with complementary skills to focus on specific scientific projects. As a PhD student, I was conducting research in the project-team named Clime.

Clime is focused on designing data assimilation methods, in the broadest sense. All methods allowing to couple observations on studied systems and numerical simulation models are investigated. Four main scientific axes are at hand:

- Estimation of the state vector characterizing a studied system, by combining observations with numerical outputs of a simulation model describing that system. The objective is to compute an analysis, which is an optimal estimation of the state vector as a compromise between the observations and the model outputs.

- Inverse modeling. The objective is to compute, for instance, sources of pollutants for an air quality or an ocean model. But inverse modeling is also applied for estimating parameters of the model or the subgrid parameters representing the phenomena that are not taken into account in the numerical model, due to the grid size.

- Uncertainty quantification and risk estimation. The approaches developed in Clime are mainly based on ensemble methods. An ensemble is designed according to the knowledge on uncertainty at initial date: uncertainty on input data, uncertainty on parameters values, uncertainty due to numerical schemes, ... This ensemble is integrated in time and allows to assess risk, according to the probability of threshold exceedance.

- Image assimilation. Two main subjects are investigated. The first one concerns the design of image processing methods in the framework of data assimilation. This includes for instance the estimation of the motion field underlying the temporal evolution displayed on an image sequence. The second one concerns the design of structured observation operators, allowing to couple a numerical model with the structures that are displayed on the image sequence.

This document will mostly concern image assimilation and will provide scientific perspectives on uncertainty quantification for motion estimation.

Images, due to the low cost of sensors, are obtained at a stunning rate in our daily life. Every one owns at least one camera. Every phone is equipped with image sensors. More and more cities are equipped with security cameras, either for pedestrian tracking, action recognition or road control.

Figure 1 – Illustration of orbiting satellites. Image Credit: European Space Agency.

Hundreds of satellites have been launched in the past 30 years and a thousand more is awaited in the next 15 years as illustrated by Figure 1. The ground based radar coverage is not neglected either for meteorological application. In France, for instance, a global coverage is achieved with the ARAMIS network, displayed on Figure 2, which is used for weather monitoring at high precision.

The always increasing information flux renders manual analysis impossible in most applications. Having that many images is a profitable evolution, as long as it will become possible to automatically process the acquisitions and the temporal sequences. The interested user, depending on its applicative goal, whether it is monitoring or forecasting, must design and implement automatic methods to extract the information he needs from the tremendous data flux he disposes.

Image analysis is a wide subject and gathers lots of applications and issues. Researchers working on photography design algorithms for automatic image classification, feature extraction or protection of copyrights. The ones working on security develop tracking methods in order to follow pedestrians or algorithms that automatically interpret scenes from everyday life. The issue of computing apparent motion from an image sequence or following structures of interest on the image data is of major importance in lots of operational systems.

Automatic methods to interpret images are also crucial in the area of weather forecast. For instance, Numerical Weather Prediction systems rely on deterministic evolution models and obser-

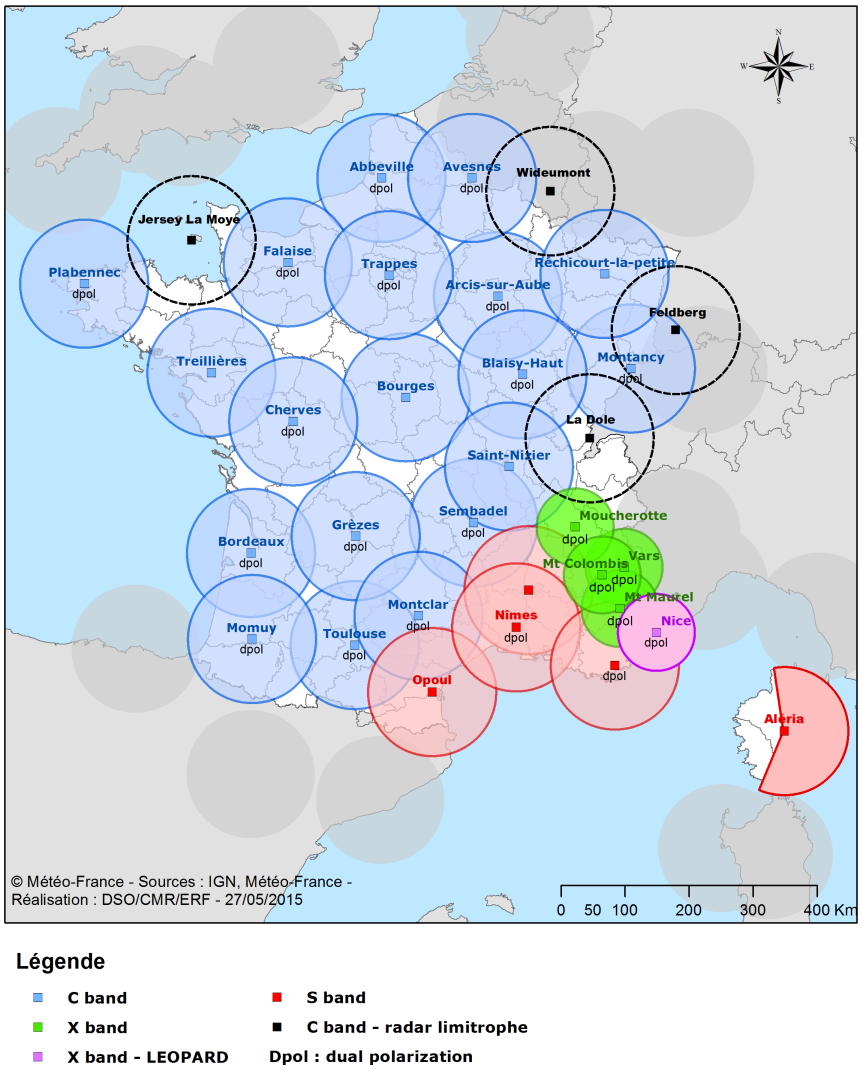

Figure 2 – Aramis network in 2015. Image Credit: Météo France.

vations, in situ measures and satellite data. The model outputs are combined to the observations with data assimilation methods in order to produce weather forecasts.

The study of the motion field underlain in an image sequence relies, in the image processing community, on the so-called optical flow algorithms, with the founding paper of Horn and Schunck [5]. These methods compute, in-between two consecutive images, the motion field that allows transforming the first image into the second one, according to heuristics on the evolution of the luminance or brightness function. However, motion estimation from image data is an ill-posed problem according to the definition given by Hadamard [6], as the luminance conservation equation, widely used in image processing, does not possess a unique solution: one equation and two unknowns. Consequently, an infinity of motion fields verify the optical flow equation and it is necessary to add information in order to render the solution unique. Smoothing of the motion field, according to the design of Tikhonov [7] regularization terms, is often used in the literature,

as seen for instance in the papers from Nagel et al. [8], Nielsen et al. [9] or more recently from Welberger et al. [10]. A huge literature is available on the subject. Survey papers on optical flow have been published, as for instance Sun et al. [11], Fortun et al. [12]. A number of databases have been created by researchers, and used for contests, in order to objectively and quantitatively compare the different methods. These databases include the ground truth of the motion field used for creating the image sequence. The Reader can for instance consider the Middlebury database, presented by Baker et al. [13], or the MPI Sintel database, introduced by Butler et al. [14].

Most methods of optical flow estimation, as the ones previously cited, have a major drawback. They do not consider the temporal consistency of the motion field: the motion function is more or less continuous in time. In most methods, after two frames have been processed, the next pair of images is treated independently. As a solution, a number of optical flow approaches, as the ones presented by Volz et al. [15] and by Vogel et al. [16] do consider the whole sequence of image acquisitions by adding temporal regularization terms in the cost function to be minimized for the estimation. However, the resulting estimation does not rely on a model of the temporal evolution of motion and does not verify the available physical knowledge on the observed system.

Having analyzed the literature, our conclusion was that heuristics on the temporal evolution of the observed system, or even mathematical laws of that evolution, written as partial differential equations, have to be included in the estimation process for two major reasons. First, these temporal equations would render the motion estimation problem well-posed and allow to avoid additional regularization methods or minimize their impact, as they often smooths too much the result. Second, these laws provide useful information on the dynamics of the studied application and as such allows to better the estimation.

Investing atmospheric or oceanic applications based on satellite acquisitions, we showed that the brightness conservation assumption is sometimes too restrictive, or even clearly not valid. Inspired by the scientific community working on structures tracking, we concluded that additional information, coded on image sequences, has to be considered for improving the motion estimation. Structures tracking is a widely studied problem in the computer vision theory. A detailed description of the state-of-the-art is given, for instance, by the survey of Yilmaz et al. in [17] and, more recently, the one done by Smeuler et al. [18].

Structures representations are numerous in the literature and they rely on different criteria. For instance, in pedestrian tracking, people can be represented with a simplified skeleton, whose points are located on major articulations and linked by straight lines. They can also be described with their bounding box, whose height/width ratio should correspond to a statistical human size ratio. Therefore, depending on operational systems, tracking algorithms rely on a different coding of what is displayed by the image sequence.

Having defined the appropriate structures representation, most of the state-of-the-art algorithms work in three steps. First, an optical flow method is applied on the image sequence in order to roughly estimate the structures displacement. Then, a second step is applied during which the structures representation is moved from the first frame to the second one, producing an estimated position of the structures on that frame. Last, in a neighborhood of that estimated position, a correction process is applied to fit the chosen representation of the structures on the second frame. The approaches described by Peterfreund et al. [19], Rathi et al. [20] and Avenel et al. [21], all enter in this processing framework. However, to our knowledge, none of these tracking techniques uses the result of the tracking algorithm to better the motion estimation.

The main idea, from which originates this PhD manuscript, is to design motion estimation

methods in which models of the structures, displayed on the image sequence, are used for obtaining a better result of motion estimation.

The mathematical approach used in the document is data assimilation, or, to be more precise, image assimilation. Data assimilation relies on a dynamic model of the studied system and on observations of that system. From these two types of information, model and data, data assimilation methods compute an analysis, which can be seen as a compromise between the model outputs and the observations. A well-known reference on data assimilation is the publication of Bouttier and Courtier [22]. Estimating motion fields with data assimilation techniques has appeared approximately 7-8 years ago. The Reader can consider for example the methods presented by Papadakis et al. [23], Béréziat et al. [1] or Ridal et al [24]. In the last few years, a number of such methods were defined for various applications.

The data assimilation technique that is applied has to be chosen according to the characteristics of the observed system. Data assimilation algorithms are divided into two main groups.

- The first one concerns the sequential data assimilation methods, as, for instance, the Kalman filter [25]. A well known, and intensively applied, sequential method is the Ensemble Kalman Filter (EnKF) defined by Evensen [4]. Sequential methods work as follows. First, starting from a background value, corresponding from the a priori knowledge available on the studied system, a simulation model is integrated in time until an acquisition date. Second, at that acquisition date, the model output is compared to the observations. An analysis is computed that results from an optimal coupling of the model output and the observations. The first step is then iterated until the next acquisition date, an so on and so forth. Therefore, at each acquisition date, the best estimation according to the acquisitions and the model output is produced, which leads to a piecewise continuous estimation. The process is illustrated on a scalar state vector $\mathbf{X}$, depending on time, by Figure 3.

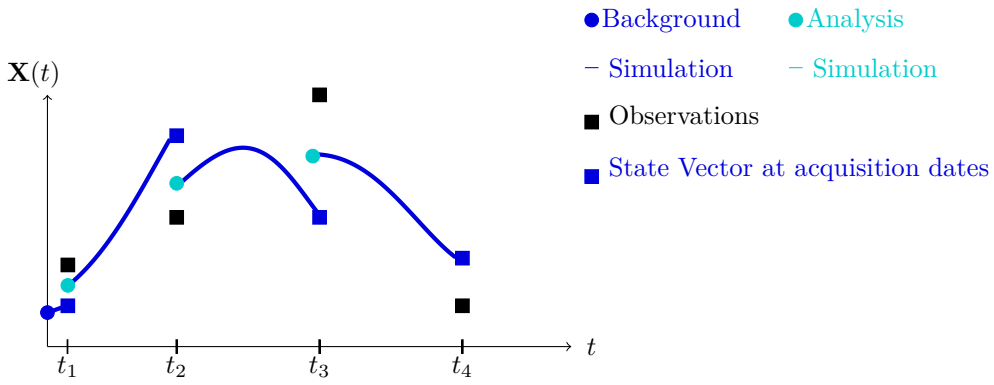

Figure 3 – Illustration of the sequential data assimilation on a scalar state vector $\mathbf{X}$.

- The second group concerns the variational data assimilation methods, based on control theory. The paper of Le Dimet and Talagrand [26] describes the solution of the 4D-Var data assimilation algorithm, thanks to the adjoint method.

  A 4D-Var data assimilation works as follows. Starting from a background value, a simulation model is integrated in time, producing a state vector value at each time step of the studied temporal interval. At each acquisition date, the state vector is compared to the observations.

The data assimilation method aims to minimize their difference on the whole temporal interval and computes the corresponding initial value, named the analysis. The final result is simultaneously that analysis and the whole trajectory resulting from its integration by the model. Figure 4 illustrates the behavior of the variational data assimilation, applied on a scalar state vector $\mathbf{X}$ that depends on time t.

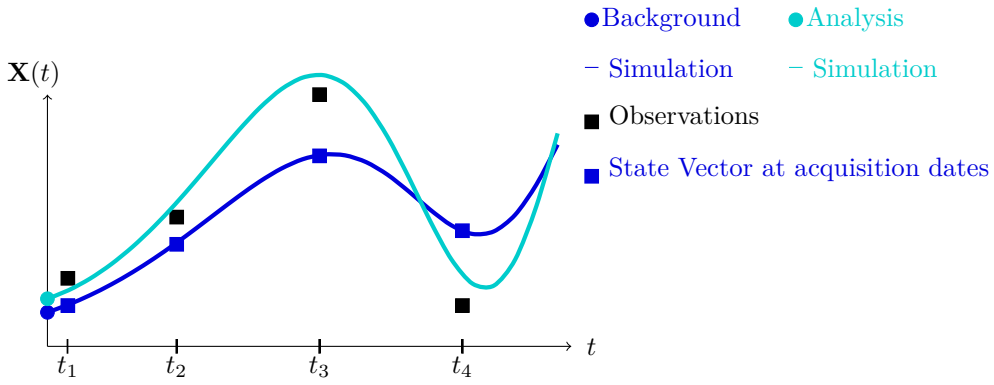

Figure 4 – Illustration of the variational data assimilation on a scalar state vector $\mathbf{X}$.

Each data assimilation method has its own advantages and drawbacks, which will be further discussed in this manuscript. Their main difference concerning the resulting motion field is that sequential filters estimate a piecewise continuous motion function, while variational methods estimate a continuous motion field that satisfies the mathematical equations described by the model. Both approaches have been applied in the context of this PhD thesis. Part I focuses on variational data assimilation, while Part II concerns the sequential approach. A description of the chapters content for these two Parts is given in the following.

- Chapter 1 describes a motion estimation method, whose input is the gray level function corresponding to the image acquisitions. That function is defined on the image domain and on the studied temporal interval.

  A detailed description of the notations that are used in Part I is given in Section 1.1.

  The Image Model, which describes the temporal evolution of the state vector, is discussed in Section 1.2. This model $\mathbb{M}$ is used to propagate in time the state vector $\mathbf{X}$, from its initial condition. Going from the continuous mathematical equations to the discrete numerical model is obtained with two types of discretization schemes, which are described in Section 1.2.

  As the motion estimation methods of Part I rely on a 4D-Var algorithm, a general description of this technique is given in Section 1.3.

  Starting from this theoretical description, each component is then adapted to the issue of motion estimation from an image sequence. The state vector, its variables, the observations and the errors on model, background and observations are defined accordingly to the image acquisitions and to their physical and mathematical properties. The 4D-Var technique is a framework in which a cost function J, depending of the model outputs and on the image acquisitions, has to be minimized. For applying a gradient descent technique, the gradient of J is mathematically calculated in Subsection 1.3.1.

However, if the cost function is only defined from the model outputs and the image acquisitions, it usually fails to estimate a smooth motion field. Consequently, it is required to add regularization terms in the definition of J. A full description of these regularization terms, of the mathematical computation of their gradient and of their impact regarding the estimation of motion is given in Section 1.4, which concludes the mathematical description of motion estimation from an image sequence.

Sections 1.5 and 1.6 describe implementation issues, the solution that has been developed in Clime and the minimization algorithm, which is used by our method. The chapter ends with a brief conclusion regarding this method of motion estimation from an image sequence.

- In some applications, and in meteorology for instance, an image sequence shows the displacement of structures, such as clouds. However, computing the motion field from the image sequence provides a partial estimation, as motion is only derived in regions where clouds are present. Therefore, approaches relying on the structures, which are displayed on an image sequence, are necessary to compute a spatially smooth motion field. This is the subject, which is discussed in Chapter 2. We designed a method, whose inputs are the image acquisitions and a representation of the structures that are displayed on these data, which estimates motion with an Image Assimilation method, based on a 4D-Var technique.

  Section 2.1 explains how to describe the structures, which are used in the estimation. The mathematical characterization is the map of the signed distance to the structures boundary. The properties of such representation are analyzed. This distance map is then included in the vector $\mathbf{X}$ representing the state of the studied system. During the assimilation process, the state vector $\mathbf{X}$ is compared with the observation vector $\mathbf{Y}$, in order to minimize their discrepancy. The observation vector is then also expanded with a new variable, compatible with the distance map. This issue is explained in Section 2.2.

  The previous modifications of vectors $\mathbf{X}$ and $\mathbf{Y}$ imply to define a new Image Model, which is used for integrating in time the structures representation. This extended model is given in Section 2.3. Moreover, some of the components of the image assimilation method, which were presented in Chapter 1, have to be redefined, as explained in the Section 2.4.

  Expanding the vectors $\mathbf{X}$ and $\mathbf{Y}$ in order to include structures representation strongly impacts the estimated motion fields. Section 2.5 analyses the results and demonstrates the contribution of the structures on the estimation. The computational issues and the implementation details are also discussed in the same section.

  Last, a brief conclusion ends the chapter and discusses on other possible representations of structures, according to the application domain.

- Expanding the state vector $\mathbf{X}$ and the observation vector $\mathbf{Y}$ imply the modifications described in Chapter 2. This can be costly and unaffordable for operational applications. The method proposed in Chapter 3 relies on the interpretation of the regularization terms presented in Chapter 1. It is possible to interpret the regularization as a model of the background error covariance matrix. The idea for making use of structures is then to correctly model the covariances in directions depending on the structure boundaries. Such a model of the background error covariance matrix would act on the estimation similarly to the signed distance map previously added in the state vector $\mathbf{X}$. The mathematical components are discussed in Section 3.1 and an analysis of the computational cost is given in Section 3.2. The motion

estimation method would then benefit from the low computational cost of the covariance modeling and gain information from the visible structures during the assimilation process through a structured regularization term.

Section 3.3 explains how to model the background error covariance matrix using structured regularization terms. This section also demonstrates why this approach gives wider possibilities compared to the one of expanding the state vector with a distance map.

- Chapter 4 is dedicated to experiments and discussions. The validation of motion estimation on real data is often not at hand due to the non availability of any ground truth. Designing twin experiments is then a good starting point for, first, validating the implementation and, second, for quantifying the methods compared to those of the state-of-the art.

  Section 4.1 describes how such twin experiments are designed, using a given motion field and an initial image (synthetic or real data) to generate an image sequence and the ground truth on motion field. The image data are then used by the Image Assimilation methods for estimating motion. The resulting estimations are then compared to the ground truth and allow to compute statistics on the errors, such as the Root Mean Square Error criteria (RMSE). Several experiments were designed in order to put in evidence the specificities of each one of the described Image Assimilation methods.

  These twin experiments are valuable as they validate the software and allow quantifying its results. However, they do not prove the ability of the method to be suitable for operational implementation on real data. Therefore, Section 4.2 displays results that have been obtained on different experiments with real images acquired by various types of sensors. This large set of sensors acquires data with different physical properties, going from satellite meteorological acquisitions from MeteoSat Second Generation to images of traffic acquired by outdoor cameras used for security assessment. This demonstrates the broad variety of possible applications of our methods of Image Assimilation.

- Chapter 5 is dedicated to an operational implementation of the method proposed in Chapter 2. It has been done with the collaboration of the company Weather Measures[2] and concerns the precipitation nowcasting over Clermond-Ferrand. After a brief introduction to the operational context given in Section 5.1, an extensive description of the application is presented in Section 5.2. The same section presents the ground radar acquisitions that are used for estimating motion. Some difficulties linked to the radar data and the various pre-processing are discussed. This is followed by a description of the operational implementation, based on a sliding window approach. This approach is widely used in image processing in case of long term image sequences. It ensures the temporal coherency of the estimations.

  In this operational application, the estimation is one important result, but not the final objective. A forecast model has to be designed for computing the nowcast of precipitation rate. This model, its discretization and properties are described in Section 5.3.

  The company Weather Measures provided several hundreds of radar acquisitions of precipitation rate, on which the algorithms have been tested. Results are analyzed in Section 5.4 and followed by some opening remarks and a description of future work in Section 5.5.

[2] `http://www.weather-measures.fr/`

- The 4D-Var data assimilation technique is an expensive method and presents some major restrictions:
  - an adjoint model is required, which needs additional theoretical work and software development,
  - the processing of an image sequence requires a number of forward integrations of the model and backward integrations of its adjoint during the optimization process. Even if the code is parallel, the whole process necessitates more computational time than a single integration of the model, as done with a filtering approach.
  - this technique provides an estimate of the motion field on the whole time interval, from the image sequence, but no uncertainty measure is at hand. That uncertainty value is however mandatory for further interpretation of motion results.

  The design of filtering methods, and in particular the ones based on the Ensemble Kalman Filter (EnKF), is an alternative that is not affected by the previous limitations and has therefore been studied in Part II of this thesis.

  EnKF is an adaptation of the Kalman filter [25] that does not use an analytical description of the probability density function but samples it. To implement the filter, an ensemble of motion fields is constructed at initial date and propagated in time.

- Chapter 6 focuses on the description of a filtering method for motion estimation considering structures in the assimilation process.

  Section 6.1, after a brief introduction on the notations, describes, starting from Bayes' formula, the behavior of the Kalman filter and its improvement by using an ensemble of state vectors.

  The general description of the sequential data assimilation algorithm is followed, in Section 6.2, by a brief recall of the Image Model, the observations and the corresponding operators. Then, the construction of the initial ensemble is described. This is done by using a broad number of optical flow algorithms from the state-of-the-art. These algorithms estimate motion in-between the two first images of the studied sequence. The whole set of results is analyzed in order to estimate its standard deviation, which is further used for randomly constructing the ensemble members.

  Section 6.3 explains why it is necessary to use an ensemble of observations in order to avoid a premature shrinking of the ensemble.

  EnKF suffers from the sampling error in the description of the background error covariance matrix $B^{(b)}$, which usually brings spurious covariance values for long distant pixels. In order to limit the impact of such covariances, a localization function, relying on the distance but also on the similarity between pixels of the image, is defined in Section 6.4. It is through this localization function that the structures visible on the image sequence are taken into account during the minimization. This chapter ends with an analysis of the computational cost of the proposed localization technique in Section 6.5.

- However, localization as presented in this Chapter 6 drastically increases the computational cost. An other localization approach has been tested, which is based on domain decomposition. Chapter 7 describes how, by decomposing the domain into subdomains and applying

EnKF on each subdomain independently from the others, it is possible to obtain a localization method that gives similar results than the one presented in Chapter 6.

In order to illustrate the principles of the localization with domain decomposition, a simplified decomposition is proposed in Section 7.1.

However, this simplified decomposition generates a discontinuous motion estimation. An alternative, allowing a smooth estimation is then designed. This last method is highly parallelizable, as each subdomain is processed independently, but suffers from the approximations done for matrix inversion, which are described in Subsection 7.1.3.

In order to mimic as much as possible the explicit localization method, a new domain decomposition is designed that relies on two parameters. The first one corresponds to a decorrelation distance and the second one describes the similarities between pixels. This structure oriented domain decomposition is described in Section 7.2 and is based on a split and merge approach.

- Part II ends with Chapter 8 on experimental results of motion estimation with the whole set of proposed sequential algorithms. The explicit localization method and the methods based on domain decomposition are tested on twin experiments in Section 8.1. Particularities of each approach are put in evidence and quantified.

  These synthetic experiments are then followed by application on real acquisitions in Section 8.2.

A conclusion chapter ends that thesis document. It summarizes the main results and gives insights for improvement of the methods and their adaptation in operational contexts.

Part I

Variationnal data assimilation for motion estimation with structure components

Chapter 1

Motion estimation with a variational data assimilation method

This chapter discusses the issue of motion estimation from image sequences, based on an image assimilation process. The goal is to compute the estimation as an optimal compromise between the image acquisitions and a physical knowledge of the observed system, expressed by an evolution model.

In order to estimate motion, we propose to use the 4D-Var method, which is a variational data assimilation technique. The estimation is achieved by minimizing a cost function, controlled by the motion field at the beginning of studied temporal interval. The method is iterative and, at each iteration, the evolution model propagates the state vector $\mathbf{X}(0)$ over the studied temporal interval, in order to compute the value $\mathbf{X}(t)$, at any date t. The state vector $\mathbf{X}(t)$ is then compared to the observations $\mathbf{Y}$ at every acquisition date. Then, using a minimization algorithm, the discrepancy between them is reduced for the whole temporal interval at once.

Section 1.1 defines the mathematical notations used to describe the variational estimation method. In the following chapters, additional notations will be given when required.

Section 1.2 describes the Image Model ruling the temporal evolution of the state vector. The discretization schemes of this model are also discussed for a better understanding of the implementation.

The image assimilation method is extensively described in Section 1.3. The method relies on the minimization of a cost function J, which requires the gradient of this function J. Calculation of the gradient is discussed in Subsection 1.3.1.

The minimization process is often unstable and leads to physically unacceptable motion results. Therefore, regularization terms are added to the cost function, as described in Section 1.4. In addition, Subsection 1.4.1 gives an interpretation of the utilization of regularization terms.

The description of the estimation process ends with details on the iterative minimization process, in Section 1.5, and a brief pseudo-code of the implementation in Section 1.6.

Last, a conclusion to the chapter, with description of possible improvements, is given in Section 1.7

1.1 Mathematical settings

- Let Ω be an open subset of $\mathbb{R}^2$, whose boundary is denoted $\partial\Omega$. Ω is the image domain on which motion is estimated.
- Let $[0,T]$ be a closed interval of $\mathbb{R}$, corresponding to the time interval on which image acquisitions are available.
- Then, $\Omega_T := \Omega \times [0,T]$ defines the studied spatio-temporal interval, on which image assimilation is applied.
- Let define $\langle \cdot \ , \ \cdot \rangle$, denoting the scalar product associated to a norm $\|.\|_H^2$ in the Hilbert space H.
- $L_2(\Omega)$ denotes the function space of f: $\mathbb{R}^2 \to \mathbb{R}$ such that:

$$||f||_{L_2}^2 := \int_\Omega f^2(x,y)\mathrm{d}x\mathrm{d}y < +\infty \tag{1.1}$$

- Let denote $\partial_x = \dfrac{\partial}{\partial x}$ and $\partial_y = \dfrac{\partial}{\partial y}$ the partial derivatives according, respectively, to the directions along x and y.
- Let define $H^m(\Omega)$ the Sobolev space of functions f: $\mathbb{R}^2 \to \mathbb{R}$, on which the following norm $\|.\|_{H^m}^2$ is defined by:

$$||f||_{H^m}^2 := \sum_{n_1+n_2 \leq m} ||\partial_x^{n_1}\partial_y^{n_2} f||_{L_2}^2 \tag{1.2}$$

- Let also define $L_2(0,T,H)$ the space of functions f: $[0,T] \to H$, verifying:

$$||f||^2 := \int_0^T ||f(t)||_H^2 \mathrm{d}t < +\infty \tag{1.3}$$

- A point of the image domain Ω is denoted by:

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.4}$$

  with x and y corresponding respectively to the abscissa and the ordinate, in a Cartesian system defined on Ω.
- Let denote $\mathbf{w}$ the motion function, defined on Ω_T, such that:

$$\mathbf{w}(\mathbf{x},t) = \begin{pmatrix} u(\mathbf{x},t) \\ v(\mathbf{x},t) \end{pmatrix} \tag{1.5}$$

  with u and v quantifying respectively the components values of motion along the abscissa and the ordinate.
- An image function I is defined on Ω_T, with the same physical properties as the image acquisitions. I is supposed to be transported by the motion function $\mathbf{w}$. Consequently, this image function corresponds to a passive tracer of the motion function.

- Let introduce $\mathbf{X}$ denoting the state vector of the observed system, depending on $\mathbf{x}$ and t and defined in Ω_T by:

$$\mathbf{X}(\mathbf{x},t) = \begin{pmatrix} \mathbf{w}(\mathbf{x},t) \\ I(\mathbf{x},t) \end{pmatrix} \tag{1.6}$$

- Data assimilation methods compare a model output with observed values of the studied system. The observation vector $\mathbf{Y}(\mathbf{x},t)$ is defined on Ω_T. Its components correspond to image acquisitions or to image features computed on these acquisitions. They are denoted by using the superscript $\cdot^O$. For instance, the image acquisitions are denoted $I^O(\mathbf{x},t)$.

- When needed, the image domain Ω is discretized but is still denoted with the same symbol, for sake of simplicity. The discrete indexes corresponding to x and y are i and j.
  Let denote $\mathbf{x}$ a point of the continuous domain and $\mathbf{p}$ the corresponding pixel of the discrete image domain.
  The number of pixels is N_x in the horizontal direction and N_y in the vertical direction.
  The vector expressing the values of one variable on the discrete image domain is then composed of $\mathrm{N}_\Omega = \mathrm{N}_x\mathrm{N}_y$ components. Let denote $\mathrm{N}_\mathbf{X}$ the number of components in the state vector $\mathbf{X}$ and $\mathrm{N}_\mathbf{Y}$ the number of components in the observation vector $\mathbf{Y}$.
  The notation of the continuous time variable t is kept for the discrete time index.

- On the discrete domain Ω, a pixel is either denoted by its row and column indexes, i and j, if it is required to identify the horizontal and vertical positions, or by the symbol $\mathbf{p}$, if its global location on the image is considered.

- For simplifying notations when discussing discretization schemes, the discrete function u at date t and pixel $\mathbf{p} = (i,j)$ is denoted:

$$u(\mathbf{p},t) = u(i,j,t) = u_t^\mathbf{p} = u_t^{ij} \tag{1.7}$$

- Discussions will consider as transparent going from the two dimensional representation of a discrete image to its storage and processing as a vector. Using one of the following two equations, it is possible to compute $\mathbf{p}$, and therefore the position in the vector, of the pixel of indexes (i,j):

  - if the image is stored row-wise:

$$\mathbf{p} \quad = \quad i + j \times \mathrm{N}_x \tag{1.8}$$

  - if the image is stored column-wise:

$$\mathbf{p} \quad = \quad j + i \times \mathrm{N}_y \tag{1.9}$$

- When describing a data assimilation method, projection operators are needed that are denoted $\mathbb{P}$. For instance, a projection from the state vector space into the space of the u component is denoted $\mathbb{P}_u$. Equivalently $\mathbb{P}_\mathbf{w}$ is the projection from the state vector space on the space of motion fields. Mathematically, the projection operators are block matrices, where some blocks are equal to the identity matrix and the other blocks are null matrices.

- When defining the formulation of the optimal estimation, error terms, denoted $\mathcal{E}$, are needed. These error terms will be considered as Gaussian. They are therefore described by a mean value and a covariance function. Mean values mostly being null, only the covariance function is needed to fully describe these error terms. Let consider the covariance function B, the corresponding error term is denoted $\mathcal{E}_B$. In discrete cases, the covariance function becomes a covariance matrix and will keep the same notation, B.

The image function I and both components, u and v, of the motion field $\mathbf{w}$ are defined on $L_2(0,T,H^1(\Omega))$, as well as any component of the observation vector $\mathbf{Y}$. For sake of simplicity, we denote u the function in $L_2([0,T],H^1(\Omega))$, $u(t)$ the field at date t belonging to $H^1(\Omega)$ and $u(\mathbf{x},t)$ the value at pixel $\mathbf{x}$ of the image domain Ω.

Having defined most of the notations used in the thesis, we arrive to the core subject of motion estimation. The next Section 1.2 will focus on the Image Model $\mathbb{IM}$ that is used to make the estimation compatible with physical assumptions on the observed system.

1.2 Image Model for motion estimation

This section describes the Image Model, $\mathbb{IM}$ that is used by the image assimilation method for estimating the motion function displayed by an image sequence.

Different heuristics are considered in the Image Model, depending on the type of image acquisitions and of their physical properties, and depending on the duration of the studied temporal interval. On a short duration, the motion field is, for instance, usually considered as stationary, which is mathematically written as:

$$\frac{\partial \mathbf{w}}{\partial t} = 0 \tag{1.10}$$

Such simple evolution law has a great potential on operational applications, as no temporal integration of the motion field is required: $\mathbf{w}(t) = \mathbf{w}(0)$ for each value of t. On longer durations, this assumption is no more valid and has to be released. Consequently, an evolution law has to be defined that rules the temporal integration of the motion field. In this document, motion is considered as advected by itself, in such cases. This assumption is mathematically written as:

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{w} \cdot \nabla \mathbf{w} = 0 \tag{1.11}$$

It corresponds to the Lagrangian conservation of motion:

$$\frac{d\mathbf{w}}{dt} = 0 \tag{1.12}$$

The motion value $\mathbf{w}(\mathbf{x},t)$ of the point $\mathbf{x}$ is then constant on its whole trajectory. Expressing the motion field $\mathbf{w}$ with its two components u and v gives:

$$\mathbf{w} = \begin{pmatrix} u \\ v \end{pmatrix} \tag{1.13}$$

and allows to decompose Equation (1.11) in two partial differential equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = 0 \tag{1.14}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = 0 \tag{1.15}$$

It should be noted that similar equations would be written if studying not only two-dimensional image data but also three dimensional acquisitions.

Considering the hypothesis that the image brightness is a physical property, which is preserved over time accordingly to the displacement of structures on the image domain, leads to:

$$I(\mathbf{x} + \delta\mathbf{x}, t + \delta t) = I(\mathbf{x}, t) \tag{1.16}$$

Assuming that the displacement $\delta\mathbf{x}$ and the time interval δt are small, Equation (1.16) is developed, accordingly to Taylor series, into:

$$I(\mathbf{x} + \delta\mathbf{x}, t + \delta t) = I(\mathbf{x}, t) + \delta\mathbf{x}\frac{\partial I}{\partial \mathbf{x}} + \delta t\frac{\partial I}{\partial t} + \dots \tag{1.17}$$

From Equations (1.16) and (1.17), it comes:

$$\frac{\partial I}{\partial t} + \frac{\delta\mathbf{x}}{\delta t}\frac{\partial I}{\partial \mathbf{x}} \approx 0 \tag{1.18}$$

Therefore, the image function brightness is considered transported by the motion field, which conducts to the optical flow equation:

$$\frac{\partial I}{\partial t} + \mathbf{w} \cdot \nabla I = 0 \tag{1.19}$$

Let consider the following Cauchy's problem:

$$\begin{cases} \dfrac{\partial u}{\partial t} &= f(t, u(t)) \\ u(0) &= u_0 \end{cases} \tag{1.20}$$

Its temporal discretization, by an Eulerian scheme, rules the time integration. One can compute u at date $t + 1$ from u at date t by:

$$u_{t+1} = u_t + dt f(t, u_t) \tag{1.21}$$

Using Equations (1.10, 1.19) or (1.11, 1.19) and applying the temporal discretization scheme allows to describe the evolution of the state vector $\mathbf{X}$ by the following equation:

$$\mathbf{X}_{t+1} = \mathbb{M}(\mathbf{X}_t). \tag{1.22}$$

Integration of the evolution model $\mathbb{M}$ requires to define the spatial discretization schemes for each equation involved in the system. Two approaches are used in the document. First, an Eulerian discretization scheme is designed that is based on the splitting principle and solves separately the linear and non linear advection terms. Second, a semi-Lagrangian scheme is applied that allows to better ensure the smoothness of the advected values. These two schemes are respectively described in Subsections 1.2.1 and 1.2.2 for the model $\mathbb{M}$ composed of Equations (1.11, 1.19).

1.2.1 Eulerian discretization

In order to deal with the non linearity of Equations (1.14, 1.15), we apply a source splitting technique to these two equations, as described by Leveque in [27]. The whole discretization process is symmetric according to u and v, and to x and y. Therefore the discretization of Equation (1.14)

is the only one developed in the following. First, the evolution equation of u is decomposed in two one-dimensional equations, one non linear advection along x and one linear advection along y. Equation (1.14) is then transformed into the system of equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{1.23}$$

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial y} = 0 \tag{1.24}$$

Equation (1.23) is written into a conservative form, leading to:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{\partial u}{\partial t} + \frac{\partial G(u)}{\partial x} = 0 \tag{1.25}$$

where the flux function G is defined by:

$$G(u) = \frac{u^2}{2} \tag{1.26}$$

The non linear equation along u, Equation (1.25), is discretized using a Godunov scheme. This scheme considers the flux at each inter-pixel boundary and solves the corresponding Riemann problem. In the following, the position x and y in the continuous domain will be linked to their discrete indexes i and j. From Equation (1.25), we derive the discrete equation:

$$u_{t+1}^{ij} = u_t^{ij} - \frac{dt}{dx}(g_{god}(u_t^{ij}, u_t^{i+1j}) - g_{god}(u_t^{i-1j}, u_t^{ij})) \tag{1.27}$$

with the function $g_{god}(.,.)$ being defined by:

$$g_{god}(a,b) = \begin{cases} G(b) & \text{if} \quad a < b \leq 0 \quad \text{or} \quad 0 < b \leq a \\ G(a) & \text{if} \quad 0 \leq a < b \quad \text{or} \quad b \leq a \leq 0 \\ 0 & \text{if} \quad a < 0 < b \end{cases} \tag{1.28}$$

Having discussed Equation (1.23), it leaves, for finishing the discretization of Equation (1.14), to consider Equation (1.24). A first order upwind scheme is chosen for this equation, which is written as:

$$u_t^{ij} = u_{t-1}^{ij} - \frac{dt}{dy}\left[\max\left(v_{t-1}^{ij}, 0\right)\left(u_{t-1}^{ij} - u_{t-1}^{ij-1}\right) + \min\left(v_{t-1}^{ij}, 0\right)\left(u_{t-1}^{ij+1} - u_{t-1}^{ij}\right)\right] \tag{1.29}$$

An upwind scheme of second order has been chosen for discretizing the evolution equation of the image function, Equation (1.19). The image function will be compared to the acquired images during the assimilation process, therefore the need in precision is higher on this component of the state vector. This scheme, described along the direction x, is written as follows:

$$I_t^{ij} = I_{t-1}^{ij} - \frac{dt}{2dx}\left[\max\left(u_{t-1}^{ij}, 0\right)\left(3I_{t-1}^{ij} - 4I_{t-1}^{i-1j} + I_{t-1}^{i-2j}\right) + \min\left(u_{t-1}^{ij}, 0\right)\left(-I_{t-1}^{i+2j} + 4I_{t-1}^{i+1j} - 3I_{t-1}^{ij}\right)\right] \tag{1.30}$$

As previously, the couples of variables (u, v) and (i, j) have symmetric roles in order to write the scheme along the direction y.

This concludes the description of the Eulerian discretization schemes used in the Image Model $\mathbb{M}$. An alternative to this Eulerian discretization is given in the next section and is called a semi-Lagrangian discretization.

1.2.2 Semi-Lagrangian discretization

The second approach for integrating in time the model composed of Equations (1.11, 1.19) relies on a semi-Lagrangian scheme, as described by Robert in [28] and by Staniforth and Côté in [29]. Let consider a pixel $\mathbf{p}$ of the discrete domain Ω, at date t, with the motion value $\mathbf{w}_t^{\mathbf{p}}$ and the brightness value $I_t^{\mathbf{p}}$. To compute the value of $I_{t+1}^{\mathbf{p}}$, one has first to compute the trajectory passing trough pixel $\mathbf{p}$ at date $t+1$ and to follow this trajectory backward in time in order to find $\mathbf{p}+\alpha$, which was the position of $\mathbf{p}$ at date t. The vector $-\alpha$ therefore corresponds to the displacement of $\mathbf{p}$ over the time interval from t to $t+1$. The displacement α is computed according to the following equation:

$$\alpha = -dt \cdot \mathbf{w}(\mathbf{p},t) = -dt \begin{pmatrix} u(\mathbf{p},t) \\ v(\mathbf{p},t) \end{pmatrix} \tag{1.31}$$

This formulation assumes that the trajectory of $\mathbf{p}$ is linear from t to $t+1$. The displacement, as computed by Equation (1.31), is then only valid if the trajectory is linear or if dt is small compared to the velocity value. Once the displacement is computed, the value $I(\mathbf{p},t+1)$ is obtained by:

$$I(\mathbf{p},t+1) = \mathscr{L}\left(I(\mathbf{p}+\alpha,t)\right) \tag{1.32}$$

The operator $\mathscr{L}()$ denotes an interpolation function based on the surrounding pixels of $\mathbf{p}+\alpha$. The interpolation is mandatory as there is no reason that $\mathbf{p}+\alpha$ represents a pixel index in Ω. As shown in [29, 30], the linear interpolation is less accurate than higher order methods, but is used as it limits the computational cost.

Having defined the Image Model $\mathbb{M}$ and the various discretization methods used to integrate this model in time, it is possible to describe the assimilation process in the next section.

1.3 Image assimilation

This section will now describe the image assimilation method that provides an optimal motion estimation from image observations $\mathbf{Y}$ of the studied system. We remind that the observation vector $\mathbf{Y}$ may be different for the original image sequence on which motion is estimated. The vector $\mathbf{Y}$ may, for instance, include characteristic features computed on the image acquisition.

The motion field $\mathbf{w}$ being part of the state vector $\mathbf{X}$, the chosen approach estimates $\mathbf{X}$ with a 4D-Var algorithm based on the following three equations:

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{x},t) + \mathbb{M}(\mathbf{X})(\mathbf{x},t) = 0 \tag{1.33}$$

$$\mathbf{X}(\mathbf{x},0) = \mathbf{X}^{(b)}(\mathbf{x}) + \mathcal{E}_B(\mathbf{x}) \tag{1.34}$$

$$\mathbb{H}(\mathbf{X},\mathbf{Y})(\mathbf{x},t) = \mathcal{E}_R(\mathbf{x},t) \tag{1.35}$$

Equation (1.33) is the partial differential equation ruling the temporal evolution of $\mathbf{X}(\mathbf{x},t)$, which is determined, for any date t, from the initial value $\mathbf{X}(\mathbf{x},0)$ and the integration of the model described in Subsection 1.2. It should be noted that the evolution equation is considered without any error term: the right component is equal to zero. This consequently implies that the physics modeled by $\mathbb{M}$ perfectly describes the evolution of the system. Such a formulation is named a strong 4D-Var method. Of course, the model $\mathbb{M}$ is usually far from perfectly modeling the temporal evolution and this assumption is then released by adding a model error, leading to

the so-called weak 4D-Var formulation. A few more details on modeling the model errors are given in the conclusion, Section 1.7.

Equation (1.34) expresses the a priori knowledge, named the background value and denoted $\mathbf{X}^{(b)}(\mathbf{x})$, that is available on the state vector at initial date 0. An error term, $\mathcal{E}_B(\mathbf{x})$, is added in order to express the uncertainty on this a priori knowledge. This error term is supposed to be Gaussian and zero-mean, with a covariance function denoted B. In our applications, B does not include any covariance between the state vector components. The choice of the background value is depending on the experiment that is conducted and will be described together with the studied images. However, as the objective is to estimate the motion field from image data, no constraint will be applied for ensuring that the result stay close to the background value of motion. This motion field is only used as a starting point for the iterative minimization process. The background of the image function I is generally taken as the first image of the acquisition sequence. If $\mathbb{P}_I$ denotes the projection of the state vector $\mathbf{X}$ on the component I, Equation (1.34) is rewritten, in this context, as:

$$\mathbb{P}_I(\mathbf{X}(\mathbf{x},0)) = \mathbb{P}_I(\mathbf{X}^{(b)}(\mathbf{x})) + \mathbb{P}_I\left(\mathcal{E}_B(\mathbf{x})\right) \tag{1.36}$$

Equation (1.36) is equivalent to:

$$I(\mathbf{x},0) - I^{(b)}(\mathbf{x}) = \mathcal{E}_{B_I}(\mathbf{x}) \tag{1.37}$$

where $\mathcal{E}_{B_I}(\mathbf{x})$ is the zero-mean Gaussian error term associated to the covariance matrix B_I defined by:

$$B_I = \mathbb{P}_I B \mathbb{P}_I^T \tag{1.38}$$

Equation (1.35) is the observation equation that links the observation value to the state vector value, at each date of the studied interval. The observation vector $\mathbf{Y}(\mathbf{x},t)$ includes the observed images $I^O(\mathbf{x},t)$ that are processed to estimate motion. For the current context of motion estimation from an image sequence, $\mathbb{H}$ allows to compare the image function $I(\mathbf{x},t)$ to the image acquisitions $I^O(\mathbf{x},t)$. It is therefore denoted $\mathbb{H}_I$ and defined by:

$$\mathbb{H}_I(\mathbf{X},\mathbf{Y})(\mathbf{x},t) = I(\mathbf{x},t) - I^O(\mathbf{x},t) \tag{1.39}$$

The observation equation, Equation (1.35), then rewrites:

$$I(\mathbf{x},t) - I^O(\mathbf{x},t) = \mathcal{E}_{R_I}(\mathbf{x},t) \tag{1.40}$$

The discrepancy between the image function $I(\mathbf{x},t)$ and the acquisition $I^O(\mathbf{x},t)$ is described by the error $\mathcal{E}_{R_I}(\mathbf{x},t)$. This error term is also supposed Gaussian, zero-mean and uncorrelated with $\mathcal{E}_{B_I}$. The covariance function R_I, associated to $\mathcal{E}_{R_I}(\mathbf{x},t)$, considers no covariance between two locations. $\mathcal{E}_{R_I}(\mathbf{x},t)$ represents both the acquisition and representativity errors.

Solving System (1.33, 1.34, 1.35) is then written as the minimization of a cost function J, depending on the control variable $\mathbf{X}(0)$, leading to the general formulation:

$$\begin{aligned} J(\mathbf{X}(\mathbf{x},0)) = \frac{1}{2}\int_\Omega \left(\mathbf{X}(\mathbf{x},0) - \mathbf{X}^{(b)}(\mathbf{x})\right) B(\mathbf{x})^{-1} \left(\mathbf{X}(\mathbf{x},0) - \mathbf{X}^{(b)}(\mathbf{x})\right) \mathrm{d}\mathbf{x} \\ + \frac{1}{2}\int_{\Omega_T} \left(\mathbb{H}(\mathbf{X},\mathbf{Y})(\mathbf{x},t)\right) R(\mathbf{x},t)^{-1} \left(\mathbb{H}(\mathbf{X},\mathbf{Y})(\mathbf{x},t)\right) \mathrm{d}\mathbf{x}\mathrm{d}t \end{aligned} \tag{1.41}$$

This general formulation can be rewritten according to the error terms $\mathcal{E}_B$ and $\mathcal{E}_R$ from Equation (1.34) and Equation (1.35) as:

$$J(\mathcal{E}_B) = \frac{1}{2}\int_{\Omega} \mathcal{E}_B(\mathbf{x})B(\mathbf{x})^{-1}\mathcal{E}_B(\mathbf{x})\mathrm{d}\mathbf{x} + \frac{1}{2}\int_{\Omega_T} \mathcal{E}_R(\mathbf{x},t)R(\mathbf{x},t)^{-1}\mathcal{E}_R(\mathbf{x},t)\mathrm{d}\mathbf{x}\mathrm{d}t \tag{1.42}$$

In the particular context of this chapter that describes motion estimation from an image sequence, Equation (1.41) can be rewritten, considering Equation (1.37) and Equation (1.40) as:

$$\begin{aligned} J(\mathbf{X}(\mathbf{x},0)) = \frac{1}{2}\int_{\Omega} \left(I(\mathbf{x},0) - I^{(b)}(\mathbf{x})\right) B_I(\mathbf{x})^{-1} \left(I(\mathbf{x},0) - I^{(b)}(\mathbf{x})\right) \mathrm{d}\mathbf{x} \\ + \frac{1}{2}\int_{\Omega_T} \left(I(\mathbf{x},t) - I^{O}(\mathbf{x},t)\right) R_I(\mathbf{x},t)^{-1} \left(I(\mathbf{x},t) - I^{O}(\mathbf{x},t)\right) \mathrm{d}\mathbf{x}\mathrm{d}t \end{aligned} \tag{1.43}$$

while Equation (1.42) rewrites:

$$J(\mathcal{E}_{B_I}) = \frac{1}{2}\int_{\Omega} \mathcal{E}_{B_I}(\mathbf{x})B_I(\mathbf{x})^{-1}\mathcal{E}_{B_I}(\mathbf{x})\mathrm{d}\mathbf{x} + \frac{1}{2}\int_{\Omega_T} \mathcal{E}_{R_I}(\mathbf{x},t)R_I(\mathbf{x},t)^{-1}\mathcal{E}_{R_I}(\mathbf{x},t)\mathrm{d}\mathbf{x}\mathrm{d}t \tag{1.44}$$

Minimizing the cost function J of Equation (1.42) allows the simultaneous minimization of the background and observation errors, $\mathcal{E}_B(\mathbf{x},t)$ and $\mathcal{E}_R(\mathbf{x},t)$, according to the norms defined by B and R. To minimize the cost function J, an adjoint variable λ is first defined that verifies, as described by Le Dimet and Talagrand in [26]:

$$\lambda(T) = 0 \tag{1.45}$$

$$-\frac{\partial\lambda}{\partial t} + \left(\frac{\partial\mathbb{M}}{\partial\mathbf{X}}\right)^{*}\lambda = \left(\frac{\partial\mathbb{H}}{\partial\mathbf{X}}\right)^{*} R^{-1}\mathbb{H}(\mathbf{X},\mathbf{Y}) \tag{1.46}$$

where $\left(\frac{\partial\mathbb{M}}{\partial\mathbf{X}}\right)^{*}$ and $\left(\frac{\partial\mathbb{H}}{\partial\mathbf{X}}\right)^{*}$ denote the adjoint operators of, respectively, the observation operator $\mathbb{M}$ and the model $\mathbb{H}$. The computation of these adjoints operators is described in Subsection 1.3.1. As demonstrated in Subsection 1.3.1, the general formula of the gradient of J is written, according to $\mathcal{E}_B$, as:

$$\frac{\partial J}{\partial \mathcal{E}_B}(\mathcal{E}_B) = B^{-1}\mathcal{E}_B + \lambda(0) \tag{1.47}$$

or, equivalently, according to $\mathbf{X}(0)$ as:

$$\frac{\partial J}{\partial \mathbf{X}(0)}(\mathbf{X}(0)) = B^{-1}\left(\mathbf{X}(0) - \mathbf{X}^{(b)}\right) + \lambda(0) \tag{1.48}$$

In the current context of motion estimation from an image sequence, as the background equation (1.37) only constrains the image component of the state vector, the gradient of J, Equation (1.48), rewrites:

$$\frac{\partial J}{\partial \mathbf{X}(0)}(\mathbf{X}(0)) = B_I^{-1}\left(I(0) - I^{(b)}\right) + \lambda(0) \tag{1.49}$$

1.3.1 Computing the gradient of J and the adjoint models

In order to minimize the cost function J, a quasi-Newtonian method is used that is described in [31, 32]. It is therefore necessary to compute the gradient of the function J, expressed in Equation (1.48). Let $\lambda(t)$ be the adjoint variable, computed by a backward integration using the two following equations:

$$\lambda(T) = 0 \tag{1.50}$$

$$-\frac{\partial \lambda(t)}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* \lambda(t) = \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1} \mathbb{H}(\mathbf{Y}, \mathbf{X})(t) \tag{1.51}$$

where $\left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*$ denotes the adjoint model of $\mathbb{M}$ and $\left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^*$ the one of $\mathbb{H}$. The adjoint model verifies, for all couple of variable (η, λ):

$$\left\langle \frac{\partial \mathbb{M}}{\partial \mathbf{X}} \eta \ , \ \lambda \right\rangle = \left\langle \eta \ , \ \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* \lambda \right\rangle \tag{1.52}$$

The state vector $\mathbf{X}$ and the cost function J (see Equation (1.42)) are depending on $\mathcal{E}_B$. Let denote $\delta\mathbf{X}$ and δJ the increments of $\mathbf{X}$ and J, obtained if $\mathcal{E}_B$ is incremented by a value $\delta\mathcal{E}_B$. Equation (1.34) leads to:

$$\delta\mathbf{X}(0) = \delta\mathcal{E}_B \tag{1.53}$$

Using the definition of δJ in the cost function formula, Equations (1.41) and (1.42), it comes:

$$\delta J = \left\langle \delta\mathcal{E}_B \ , \ B^{-1}\mathcal{E}_B \right\rangle + \int_{[0,T]} \left\langle \delta\mathbf{X}(t) \ , \ \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1} \mathbb{H}(\mathbf{Y}, \mathbf{X})(t) \right\rangle \tag{1.54}$$

The evolution equation of $\mathbf{X}$, Equation (1.33), gives:

$$\frac{\partial \delta\mathbf{X}(t)}{\partial t} + \frac{\partial \mathbb{M}}{\partial \mathbf{X}} \delta\mathbf{X}(t) = 0 \tag{1.55}$$

Multiplying Equation (1.55) by $\lambda(t)$, and integrating it in space and time, leads to the following equality:

$$\int_{[0,T]} \left\langle \frac{\partial \delta\mathbf{X}(t)}{\partial t} \ , \ \lambda(t) \right\rangle + \int_{[0,T]} \left\langle \frac{\partial \mathbb{M}}{\partial \mathbf{X}} \delta\mathbf{X}(t) \ , \ \lambda(t) \right\rangle = 0 \tag{1.56}$$

Using the integration by parts theorem on the first term of the Equation (1.56), and the adjoint model property, Equation (1.52), for the second term, it comes:

$$-\int_{[0,T]} \left\langle \delta\mathbf{X}(t) \ , \ \frac{\partial \lambda(t)}{\partial t} \right\rangle + \langle \delta\mathbf{X}(T) \ , \ \lambda(T) \rangle - \langle \delta\mathcal{E}_B \ , \ \lambda(0) \rangle + \int_{[0,T]} \left\langle \delta\mathbf{X}(t) \ , \ \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* \lambda(t) \right\rangle = 0 \tag{1.57}$$

Equation (1.50) shows that $\langle \delta\mathbf{X}(T) \ , \ \lambda(T) \rangle$ is null.
Multiplying Equation (1.51) by $\delta\mathbf{X}(t)$ and integrating it in space and time leads to:

$$-\left\langle \delta\mathbf{X}(t) \ , \ \frac{\partial \lambda(t)}{\partial t} \right\rangle + \left\langle \delta\mathbf{X}(t) \ , \ \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* \lambda(t) \right\rangle = \left\langle \delta\mathbf{X}(t) \ , \ \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1} \mathbb{H}(\mathbf{Y}, \mathbf{X})(t) \right\rangle \tag{1.58}$$

Equation (1.58) allows to rewrite Equation (1.57) as:

$$\int_{[0,T]} \left\langle \delta\mathbf{X}(t) \ , \ \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1} \mathbb{H}(\mathbf{Y}, \mathbf{X})(t) \right\rangle = \langle \delta\mathcal{E}_B \ , \ \lambda(0) \rangle \tag{1.59}$$

Introducing Equation (1.59) into the second part of Equation (1.54) gives:

$$\delta J = \left\langle \delta\mathcal{E}_B \ , \ B^{-1}\mathcal{E}_B \right\rangle + \langle \delta\mathcal{E}_B \ , \ \lambda(0) \rangle \tag{1.60}$$

which finally leads to the value of the gradient of J:

$$\frac{\partial J}{\partial \mathcal{E}_B}(\mathcal{E}_B) = B^{-1}\mathcal{E}_B + \lambda(0) \tag{1.61}$$

which writes, according to $\mathbf{X}(0)$, as:

$$\frac{\partial J}{\partial \mathbf{X}(0)}(\mathbf{X}(0)) = B^{-1}\left(\mathbf{X}(0) - \mathbf{X}^{(b)}\right) + \lambda(0) \tag{1.62}$$

Equation (1.61) and Equation (1.103) allow to compute the gradient of the cost function and therefore allow the use of iterative minimization methods. To get the value of this gradient, it is then necessary to compute the value of $\lambda(0)$. Therefore, starting from $\lambda(T) = 0$, Equation (1.51) is integrated backward in time at each iteration of the minimization. This backward equation is based on the adjoint $\left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*$ of the Image Model $\mathbb{M}$ and on the adjoint $\left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^*$ of the observation operator $\mathbb{H}$.

Computing the adjoint of an operator can be done manually or automatically. Considering manual computation, the adjoint can either be calculated from the continuous operator and then discretized or calculated from the discretized operator. However, the discretization of a continuous adjoint operator is not equivalent to the adjoint of a discretized operator. The latter solution is then mandatory for computing the discrete adjoint in order to get accurate results.

In order to avoid having to manually code the adjoint, which is time consuming and subject to errors, the process has been automated. As described by Giering et al. in [33], the coding of the adjoint follows mathematical rules, which makes the process propitious to automatization. Some data assimilation frameworks, as the one proposed in the library Yao by Nardi et al. [34], require the operator to be coded by small independent functions. Moreover, the code of the adjoint has to be written by the user, for each of these functions. Last, a modular graph must be furnished, that represents the operator as a whole. From these inputs, the data assimilation library Yao is then able to automatically generate the adjoint graph. A second approach, which is less relying on user implementation and more automatic, consists in computing the adjoint of the whole operator. An automatic adjoint coding software, named Tapenade, developed by Hascoët et al. and described in [35], is used in the research work described in this document, in order to generate the adjoint $\left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*$ of the model $\mathbb{M}$. In the studied application, the observation operator $\mathbb{H}$ is a projection onto a part of the state vector $\mathbf{X}$. Its adjoint $\left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^*$ is therefore equal to the transpose of the

projection matrix. Having computed the adjoint model, the verification of its value has to be done by using Equation (1.52) applied on $\mathbf{X}$, leading to the following formula:

$$\left\langle \frac{\partial \mathbb{M}}{\partial \mathbf{X}} \mathbf{X} \ , \ \mathbf{X} \right\rangle = \left\langle \mathbf{X} \ , \ \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}} \right)^* \mathbf{X} \right\rangle \tag{1.63}$$

This section, by describing the calculation of the gradient of the cost function and how the adjoints are obtained, finishes the description of the data assimilation framework. We should remind that the aim of the chapter is to describe a motion estimation algorithm whose input is an image sequence. Applying the method, as it has been described at this stage, leads to irregular motion fields. In order to control the smoothness of the solution, the cost function of Equation (1.41) is expanded with three different regularization terms, described in the next section.

1.4 Regularization

To ensure that the estimation verifies some predetermined physical properties, additional terms are added to the cost function defined by Equation (1.41). In practice, these heuristics concern the smoothness characteristics that should be verified by the result. These additional terms, named Tikhonov regularizations, are even mandatory when estimating motion, if the estimation process is only relying on image data, as discussed by Sun et al. in [11] or in the survey from Baker et al. in [13]. This comes from the fact that the optical flow estimation is an ill-posed problem according to the Hadamard definition [6].

When estimating motion with a data assimilation technique, additional information about the temporal evolution of the motion field is included in the process. The problem is no more ill-posed. However, regularization terms are used, as described by Béréziat et al. in [36], in order to determine the vectorial subspace on which motion is estimated. These regularization terms ensure that, during the minimization process, the motion field keeps the chosen regularity properties.

Three different regularization terms are designed, weighted with parameters α, β and γ, and added to the cost function J. The values of α, β and γ are chosen so that the regularity terms constrain the minimizer but do not take over the background and observation terms of the cost function.

As all methods of image assimilation discussed in this thesis document are defined according to the strong formulation of data assimilation, there is no model error involved in the process. The regularization terms acting on the motion field are then only depending on its value at the initial date 0.

The first regularization term, $\mathcal{R}_1$, acts on the norm of the gradient of the motion field. It is designed as follows:

$$\mathcal{R}_1(\mathbf{X}(0)) = \frac{1}{2}\alpha \int_\Omega ||\nabla \left(\mathbb{P}_{\mathbf{w}}(\mathbf{X}(\mathbf{x}, 0)) \right)||^2 \mathrm{d}\mathbf{x} \tag{1.64}$$

This term $\mathcal{R}_1$ ensures a spatial smoothness of the estimation.

When working on sea surface circulation, due to the incompressibility property, the estimated motion field should be divergence free. In other applications, if the divergence is high, aliasing effects will appear during the temporal integration of the Image Model. A second regularization term $\mathcal{R}_2$ is then added to the cost function J. The divergence regularization term $\mathcal{R}_2$ is defined by:

$$\mathcal{R}_2(\mathbf{X}(0)) = \frac{1}{2}\beta \int_\Omega \left[\mathrm{div}(\mathbb{P}_{\mathbf{w}}(\mathbf{X}(0))) \right]^2 \mathrm{d}\mathbf{x} \tag{1.65}$$

where $\text{div}(\mathbb{P}_{\mathbf{w}}(\mathbf{X}(0)))$ is equal to:

$$div(\mathbb{P}_{\mathbf{w}}(\mathbf{X}(0))) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \tag{1.66}$$

Last, a regularization term acting on the norm of the motion field is also added, in order to avoid having spurious high values. This term $\mathcal{R}_3$ is defined by:

$$\mathcal{R}_3(\mathbf{X}(0)) = \frac{1}{2}\gamma \int_{\Omega} ||\mathbb{P}_{\mathbf{w}}(\mathbf{X}(0))||^2 \mathrm{d}\mathbf{x} \tag{1.67}$$

The three terms $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$ are added to the cost function of Equation (1.41) in order to obtain a new definition:

$$J(\mathbf{X}(0)) = \frac{1}{2}\int_{\Omega} \left(\mathbf{X}(0) - \mathbf{X}^{(b)}\right) B^{-1} \left(\mathbf{X}(0) - \mathbf{X}^{(b)}\right) \mathrm{d}\mathbf{x} + \frac{1}{2}\int_{\Omega_T} \left(\mathbb{H}(\mathbf{X}, \mathbf{Y})\right) R^{-1} \left(\mathbb{H}(\mathbf{X}, \mathbf{Y})\right) \mathrm{d}\mathbf{x}\mathrm{d}t + \mathcal{R}(\mathbf{X}(0)) \tag{1.68}$$

where:

$$\mathcal{R}(\mathbf{X}(0)) = \mathcal{R}_1(\mathbf{X}(0)) + \mathcal{R}_2(\mathbf{X}(0)) + \mathcal{R}_3(\mathbf{X}(0)) \tag{1.69}$$

that is minimized during the data assimilation process, which estimates $\mathbf{X}(0)$ and its motion component.

As explained before, the minimization algorithm requires the value of the gradient of the cost function. This implies computing the gradient of each regularization term. For that, one has to consider the definition of the directional derivative of the cost function J. The derivative, according to $\mathbf{X}(0)$, of J in the direction η, is defined by:

$$\frac{\partial J}{\partial \mathbf{X}(0)}(\eta) = \lim_{\nu \to 0} \frac{d}{d\nu} J(\mathbf{X}(0) + \nu\eta) \tag{1.70}$$

Computing $\dfrac{\partial \mathcal{R}_1}{\partial \mathbf{X}(0)}$ implies to study the term $\mathcal{R}_1(\mathbf{X}(0) + \nu\eta)$. It comes:

$$\mathcal{R}_1(\mathbf{X}(0) + \nu\eta) = \frac{1}{2}\alpha \int_{\Omega} (\nabla \mathbb{P}_{\mathbf{w}}(\mathbf{X}(\mathbf{x}, 0) + \nu\eta(\mathbf{x})))^T (\nabla \mathbb{P}\mathbf{w}(\mathbf{X}((\mathbf{x}, 0) + \nu\eta(\mathbf{x}))) \mathrm{d}\mathbf{x} \tag{1.71}$$

We first apply a Taylor decomposition to $\mathcal{R}_1(\mathbf{X}(0) + \nu\eta)$ and then derive the result according to ν. The constant terms of the Taylor decomposition will nullify with the derivation. The terms in ν^2 will be transformed in terms in ν when performing the derivation. Consequently, they will also nullify when ν will tend to zero. In conclusion, the only term that has to be considered is the one depending of ν in the Taylor decomposition. It comes, after an integration by parts:

$$\frac{\partial \mathcal{R}_1}{\partial \mathbf{X}(0)}(\eta) = -\alpha \int_{\Omega} \eta^T \mathbb{P}_{\mathbf{w}}^T \Delta \mathbb{P}_{\mathbf{w}}(\mathbf{X}(\mathbf{x}, 0)) \mathrm{d}\mathbf{x} \tag{1.72}$$

Remembering that $\mathbf{X}(0) = \begin{pmatrix} \mathbf{w}(0) \\ I(0) \end{pmatrix}$, it follows:

$$\frac{\partial \mathcal{R}_1}{\partial \mathbf{X}(0)} = \begin{pmatrix} -\alpha \Delta \mathbf{w}(0) \\ 0 \end{pmatrix} \tag{1.73}$$

The same calculations have to be done for $\mathcal{R}_2$ defined by:

$$\mathcal{R}_2(\mathbf{X}(0)) = \frac{1}{2}\beta \int_\Omega \left[\mathrm{div}(\mathbb{P}_\mathbf{w}(\mathbf{X}(\mathbf{x},0)))\right]^2 \mathrm{d}\mathbf{x} \tag{1.74}$$

Applying the calculus of variations gives:

$$\mathcal{R}_2(\mathbf{X}(0) + \nu\eta) = \frac{1}{2}\beta \int_\Omega \left[\mathrm{div}\left(\mathbb{P}_\mathbf{w}(\mathbf{X}(\mathbf{x},0) + \nu\eta(\mathbf{x}))\right)\right]^2 \mathrm{d}\mathbf{x} \tag{1.75}$$

The same rule than the one applied for computing $\dfrac{\partial \mathcal{R}_1}{\partial \mathbf{X}(0)}$ gives:

$$\frac{\partial \mathcal{R}_2}{\partial \mathbf{X}(0)}(\eta) = -\beta \int_\Omega \eta^T \mathbb{P}_\mathbf{w}^T \nabla(\mathrm{div}(\mathbb{P}_\mathbf{w}(\mathbf{X}(\mathbf{x},0)))) \mathrm{d}\mathbf{x} \tag{1.76}$$

which leads to:

$$\frac{\partial \mathcal{R}_2}{\partial \mathbf{X}(0)} = \begin{pmatrix} -\beta\nabla\,\mathrm{div}\left(\mathbf{w}(0)\right) \\ 0 \end{pmatrix} \tag{1.77}$$

Computation of $\dfrac{\partial \mathcal{R}_3}{\partial \mathbf{X}(0)}$ is straightforward and equal to:

$$\frac{\partial \mathcal{R}_3}{\partial \mathbf{X}(0)} = \begin{pmatrix} \gamma\mathbf{w}(0) \\ 0 \end{pmatrix} \tag{1.78}$$

The gradient of the cost function (1.68) is then equal to:

$$\frac{\partial J}{\partial \mathbf{X}(0)} = B^{-1}\left(\mathbf{X}(0) - \mathbf{X}^{(b)}\right) + \lambda(0) + \frac{\partial \mathcal{R}}{\partial \mathbf{X}(0)}\mathbf{X}(0) \tag{1.79}$$

In the context of motion estimation from an image sequence and with the previous definition of $\mathcal{R}(\mathbf{X}(0))$, it comes:

$$\frac{\partial J}{\partial \mathbf{X}(0)} = \lambda(0) + \begin{pmatrix} -\alpha\Delta\mathbf{w}(0) - \beta\nabla\,\mathrm{div}\left(\mathbf{w}(0)\right) + \gamma\mathbf{w}(0) \\ B_I^{-1}\left(I(0) - I^{(b)}\right) \end{pmatrix} \tag{1.80}$$

Using the expanded cost function Equation (1.68), the state vector $\mathbf{X}(0)$ that gives the minimum of J is the value that best describes the observations, verifies the model and conforms to the regularization constraints. The next subsection will further discuss the regularization terms and show a link between applying a regularization and defining a non diagonal background error covariance matrix.

1.4.1 Interpretation of the regularization terms

This section aims to define an interpretation of the regularization terms in order to better understand their impact on the results. Some related work has been done by Dean S. Oliver in [37], where the regularization terms are associated to the inverse of a covariance matrix. A mathematical demonstration will be given below to bring to light that the motion estimation, obtained with a diagonal background error covariance matrix B_I and regularization terms, is the same than the

one obtained with a non diagonal $B_{\mathcal{R}}$ and no regularization. In order to simplify the notations and keeping in mind that the regularization only acts at initial date 0, we denote, in this subsection, u, v and $\mathbf{w}$ the fields $u(0)$, $v(0)$, $\mathbf{w}(0)$ at initial date. This means that, in the following, $u(\mathbf{x})$ is the value of $u(\mathbf{x}, 0)$ (same for v and $\mathbf{w}$).

In the cost function of Equation (1.68), the regularization terms are given, with a variational formulation, by:

$$\begin{aligned} \mathcal{R}(\mathbf{X}(0)) &= \mathcal{R}_1(\mathbf{X}(0)) + \mathcal{R}_2(\mathbf{X}(0) + \mathcal{R}_3(\mathbf{X}(0)) && (1.81) \\ &= \frac{1}{2}\int_\Omega \alpha||\nabla \mathbf{w}(\mathbf{x})||_2^2 + \beta||\operatorname{div}(\mathbf{w}(\mathbf{x}))||_2^2 + \gamma||\mathbf{w}(\mathbf{x})||_2^2 \mathrm{d}\mathbf{x} && (1.82) \end{aligned}$$

Keeping in mind that $\mathbf{w} = \begin{pmatrix} u \\ v \end{pmatrix}$, Equation (1.82) is equivalent to:

$$\begin{aligned} \mathcal{R}(\mathbf{X}(0)) = \frac{1}{2}\int_\Omega \alpha \left(\partial_x u(\mathbf{x})^2 + \partial_y u(\mathbf{x})^2 + \partial_x v(\mathbf{x})^2 + \partial_y v(\mathbf{x})^2\right) \\ + \beta\left(\partial_x u(\mathbf{x}) + \partial_y v(\mathbf{x})\right)^2 + \gamma\left(u(\mathbf{x})^2 + v(\mathbf{x})^2\right) \mathrm{d}\mathbf{x} \quad (1.83) \end{aligned}$$

When implementing the method on the image domain Ω, the derivatives along x and y are computed by filters $\mathcal{D}_\chi$ and $\mathcal{D}_y$, which depend on the chosen discretization schemes. If the derivative along x is, for instance, approximated with a forward scheme, the filter $\mathcal{D}_\chi$ is defined by:

$$\mathcal{D}_\chi = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{-1}{dx} & \frac{1}{dx} \\ 0 & 0 & 0 \end{bmatrix} \quad (1.84)$$

The derivative filters being applied on the whole domain, let introduce the matrices D_x and D_y, which compute the discrete derivatives at every pixel, respectively along the directions x and y. By definition, D_x and D_y are Toeplitz matrices and their coefficients along descending diagonals are constant. For instance, D_x has the value $\frac{-1}{dx}$ on its main diagonal and the value $\frac{1}{dx}$ on the first above diagonal. The issue of computing gradient values on the boundary of the domain is assessed by duplicating the pixels. Equation (1.83) is rewritten in a discrete formulation with these new notations:

$$\begin{aligned} \mathcal{R}(\mathbf{X}(0)) = \frac{1}{2}\left[\alpha\left(\langle D_x u\ ,\ D_x u\rangle + \langle D_y u\ ,\ D_y u\rangle + \langle D_x v\ ,\ D_x v\rangle + \langle D_y v\ ,\ D_y v\rangle\right)\right. \\ \left. +\beta\left(\langle D_x u + D_y v\ ,\ D_x u + D_y v\rangle\right) + \gamma\left(\langle u\ ,\ u\rangle + \langle v\ ,\ v\rangle\right)\right] \quad (1.85) \end{aligned}$$

Introducing the vector $\begin{pmatrix} u \\ v \end{pmatrix}$ in each scalar product leads to:

$$\begin{aligned} \mathcal{R}(\mathbf{X}(0)) = \frac{1}{2}\left[\alpha\left(\left\langle \begin{pmatrix} D_x & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\ ,\ \begin{pmatrix} D_x & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\right\rangle + \left\langle \begin{pmatrix} D_y & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\ ,\ \begin{pmatrix} D_y & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\right\rangle \right. \\ \left. + \left\langle \begin{pmatrix} 0 & D_x \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\ ,\ \begin{pmatrix} 0 & D_x \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\right\rangle + \left\langle \begin{pmatrix} 0 & D_y \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\ ,\ \begin{pmatrix} 0 & D_y \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\right\rangle\right) \\ \left. +\beta\left\langle \begin{pmatrix} D_x & D_y \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\ ,\ \begin{pmatrix} D_x & D_y \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}\right\rangle + \gamma\left\langle \begin{pmatrix} u \\ v \end{pmatrix}\ ,\ \begin{pmatrix} u \\ v \end{pmatrix}\right\rangle\right] \quad (1.86) \end{aligned}$$

As D_x and D_y are matrices with real coefficients, their adjoints are equal to their transpose and the previous equation is transformed in:

$$\begin{aligned}\mathcal{R}(\mathbf{X}(0)) = \frac{1}{2}\Bigg[\alpha\Bigg(&\left\langle\begin{pmatrix}u\\v\end{pmatrix}\ ,\ \begin{pmatrix}D_x^TD_x & 0\\0 & 0\end{pmatrix}\begin{pmatrix}u\\v\end{pmatrix}\right\rangle + \left\langle\begin{pmatrix}u\\v\end{pmatrix}\ ,\ \begin{pmatrix}D_y^TD_y & 0\\0 & 0\end{pmatrix}\begin{pmatrix}u\\v\end{pmatrix}\right\rangle\\ &+\left\langle\begin{pmatrix}u\\v\end{pmatrix}\ ,\ \begin{pmatrix}0 & 0\\0 & D_x^TD_x\end{pmatrix}\begin{pmatrix}u\\v\end{pmatrix}\right\rangle + \left\langle\begin{pmatrix}u\\v\end{pmatrix}\ ,\ \begin{pmatrix}0 & 0\\0 & D_y^TD_y\end{pmatrix}\begin{pmatrix}u\\v\end{pmatrix}\right\rangle\Bigg)\\ &+\beta\left\langle\begin{pmatrix}u\\v\end{pmatrix}\ ,\ \begin{pmatrix}D_x^TD_x & D_x^TD_y\\D_y^TD_x & D_y^TD_y\end{pmatrix}\begin{pmatrix}u\\v\end{pmatrix}\right\rangle + \gamma\left\langle\begin{pmatrix}u\\v\end{pmatrix}\ ,\ \begin{pmatrix}u\\v\end{pmatrix}\right\rangle\Bigg]\end{aligned} \tag{1.87}$$

Using the bilinearity of the scalar product, it comes:

$$\mathcal{R}(\mathbf{X}(0)) = \frac{1}{2}\left\langle\begin{pmatrix}u\\v\end{pmatrix}\ ,\ \begin{pmatrix}(\alpha+\beta)D_x^TD_x + \alpha D_y^TD_y + \gamma Id & \beta D_x^TD_y\\ \beta D_y^TD_x & \alpha D_x^TD_x + (\alpha+\beta)D_y^TD_y + \gamma Id\end{pmatrix}\begin{pmatrix}u\\v\end{pmatrix}\right\rangle \tag{1.88}$$

where Id denotes the identity matrix.

Let define a matrix C by:

$$C = \begin{pmatrix}(\alpha+\beta)D_x^TD_x + \alpha D_y^TD_y + \gamma Id & \beta D_x^TD_y\\ \beta D_y^TD_x & \alpha D_x^TD_x + (\alpha+\beta)D_y^TD_y + \gamma Id\end{pmatrix} \tag{1.89}$$

The regularization value defined in Equation (1.82) is either positive or null, as long as α, β and γ are positive. As Equations (1.82) and (1.88) are equivalent, it is clear that $\mathcal{R}(\mathbf{w})$ nullifies if, and only if, $\mathbf{w}$ is null. Therefore, the matrix C is symmetric definite positive and can be considered as the inverse of a covariance matrix $B_{\mathbf{w}}^{-1}$. Let recall that the discrete cost function writes:

$$\begin{aligned}J(\mathbf{X}(0)) = &\frac{1}{2}\left\langle\mathbf{X}(0) - \mathbf{X}^{(b)}\ ,\ B^{-1}\left(\mathbf{X}(0) - \mathbf{X}^{(b)}\right)\right\rangle\\ &+\frac{1}{2}\int_{[0,T]}\left\langle\mathbb{H}(\mathbf{X},\mathbf{Y})(t)\ ,\ R^{-1}\left(\mathbb{H}(\mathbf{X},\mathbf{Y})(t)\right)\right\rangle \mathrm{d}t + \mathcal{R}(\mathbf{X}(0))\end{aligned} \tag{1.90}$$

which, in the context of motion estimation from an image sequence, gives:

$$\begin{aligned}J(\mathbf{X}(0)) = &\frac{1}{2}\left\langle I(0) - I^{(b)}\ ,\ B_I^{-1}\left(I(0) - I^{(b)}\right)\right\rangle\\ &+\frac{1}{2}\sum_{[0,T]}\left\langle I(t) - I^O(t)\ ,\ R_I(t)^{-1}\left(I(t) - I^O(t)\right)\right\rangle \mathrm{d}t + \mathcal{R}(\mathbf{X}(0))\end{aligned} \tag{1.91}$$

This is equal to:

$$J(\mathbf{X}(0)) = \frac{1}{2}\left\langle\mathbf{X}(0) - \mathbf{X}^{(b)}\ ,\ B_{\mathcal{R}}^{-1}\left(\mathbf{X}(0) - \mathbf{X}^{(b)}\right)\right\rangle + \frac{1}{2}\sum_{[0,T]}\left\langle I(t) - I^O(t)\ ,\ R_I^{-1}(t)\left(I(t) - I^O(t)\right)\right\rangle \mathrm{d}t \tag{1.92}$$

where the new covariance matrix $B_{\mathcal{R}}$ verifies:

$$B_{\mathcal{R}}^{-1} = \begin{pmatrix}B_{\mathbf{w}}^{-1} & 0\\ 0 & B_I^{-1}\end{pmatrix} \tag{1.93}$$

$$= \begin{pmatrix}(\alpha+\beta)D_x^TD_x + \alpha D_y^TD_y + \gamma Id & \beta D_x^TD_y & 0\\ \beta D_y^TD_x & \alpha D_x^TD_x + (\alpha+\beta)D_y^TD_y + \gamma Id & 0\\ 0 & 0 & B_I^{-1}\end{pmatrix} \tag{1.94}$$

It should be noted that in Equation (1.91), only the image component $I^{(b)}$ of $\mathbf{X}^{(b)}$ has to be defined and is equal to the first image acquisition $I^O(0)$. In Equation (1.92), the whole background vector is involved. This background vector $\mathbf{X}^{(b)}$ is then defined by:

$$\mathbf{X}^{(b)} = \begin{pmatrix} 0 \\ 0 \\ I(0) \end{pmatrix} \tag{1.95}$$

where the image component is unchanged and the motion component is given a null value as it corresponds to the a priori knowledge of the regularity of motion. The error covariance matrix B_I on the image background keeps the same value value.

This concludes the demonstration that the use of regularization terms is equivalent to the use of a non diagonal covariance matrix B.

During the computation, the state vector $\mathbf{X}(0)$ is composed of the three components $u(0)$, $v(0)$, and $I(0)$. Each of these components is defined on the discrete image domain Ω, composed of N_Ω pixels. Therefore, $\mathbf{X}(0)$ is composed of $3\mathrm{N}_\Omega$ elements. The size of the covariance matrix B is equal to the square of the size of the state vector. This leads to unaffordable memory costs if one wants to store this whole matrix. For instance, for a 100×100 pixels image, this leads to a 54 gigabytes matrix. However, the inverse matrix designed in Equation (1.94) is sparse and contains a high number of zero values. A sparse storage of this matrix is then feasible, but it would lead to high computational costs when performing the product of the matrix by a vector, for instance in Equation (1.92).

As the blocks included in $B_{\mathcal{R}}^{-1}$ are Toeplitz matrices, the best way to compute the value of the cost function J with Equation (1.92) is to consider each block of $B_{\mathcal{R}}^{-1}$ as a filter. Let determine the filters, i.e. the local functions, needed to compute $B_{\mathcal{R}}^{-1}$ during the assimilation process, without storing a dense matrix. Let remark that the filter associated to B_I^{-1} is defined by:

$$\mathcal{B}_I^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_I} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.96}$$

For further illustrating the discussion, let consider that the derivatives are computed with forward schemes, which are determined by the following convolution filters:

$$\mathcal{D}_\mathcal{X} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{-1}{dx} & \frac{1}{dx} \\ 0 & 0 & 0 \end{bmatrix}, \qquad \mathcal{D}_\mathcal{Y} = \begin{bmatrix} 0 & \frac{1}{dy} & 0 \\ 0 & \frac{-1}{dy} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.97}$$

Let denote $B_{\mathcal{R}_{i,j}}^{-1}$ the bloc on the i^{th} line and j^{th} column of $B_{\mathcal{R}}^{-1}$, as it is written in Equation (1.94). Let denote $\mathcal{B}_{\mathcal{R}_{i,j}}^{-1}$ the corresponding convolution filter. Using the mathematical rules for addition

and composition of filters, it comes:

$$\mathcal{B}^{-1}_{\mathcal{R}_{1,1}} = \begin{bmatrix} 0 & -\alpha & 0 \\ -(\alpha+\beta) & 2\times\dfrac{\alpha+\beta}{dx^2} + 2\times\dfrac{\beta}{dy^2} + \gamma & -(\alpha+\beta) \\ 0 & -\alpha & 0 \end{bmatrix} \tag{1.98}$$

$$\mathcal{B}^{-1}_{\mathcal{R}_{2,2}} = \begin{bmatrix} 0 & -(\alpha+\beta) & 0 \\ -\alpha & 2\times\dfrac{\beta}{dx^2} + 2\times\dfrac{\alpha+\beta}{dy^2} + \gamma & -\alpha \\ 0 & -(\alpha+\beta) & 0 \end{bmatrix} \tag{1.99}$$

$$\mathcal{B}^{-1}_{\mathcal{R}_{1,2}} = \begin{bmatrix} \dfrac{\beta}{dxdy} & \dfrac{-\beta}{dxdy} & 0 \\ \dfrac{-\beta}{dxdy} & \dfrac{\beta}{dxdy} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.100}$$

$$\mathcal{B}^{-1}_{\mathcal{R}_{2,1}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \dfrac{\beta}{dxdy} & \dfrac{-\beta}{dxdy} \\ 0 & \dfrac{-\beta}{dxdy} & \dfrac{\beta}{dxdy} \end{bmatrix} \tag{1.101}$$

Therefore, it is possible to model $B^{-1}_{\mathcal{R}}$ by only considering the above filters.

The design of a non diagonal matrix for the background covariance $B_{\mathcal{R}}$ allows, as demonstrated above, to apply a regularization of the state vector. During the implementation phase, it provides the advantage that the derivatives of the regularization functions defined by Equations (1.73, 1.77, 1.78) are no more required. Moreover, the filters included in the matrix $B^{-1}_{\mathcal{R}}$ are applied both in the forward integration, computing the cost function J:

$$J(\mathbf{X}(0)) = \frac{1}{2}\left\langle \mathbf{X}(0) - \mathbf{X}^{(b)} \ , \ B^{-1}_{\mathcal{R}}\left(\mathbf{X}(0) - \mathbf{X}^{(b)}\right)\right\rangle + \frac{1}{2}\sum_{[0,T]}\left\langle I(t) - I^{O}(t) \ , \ R_I(\mathbf{x},t)^{-1}\left(I(t) - I^{O}(t)\right)\right\rangle \mathrm{d}t \tag{1.102}$$

and in the backward integration, computing its gradient $\dfrac{\partial J}{\partial \mathbf{X}(0)}$:

$$\frac{\partial J}{\partial \mathbf{X}(0)}(\mathbf{X}(0)) = B^{-1}_{\mathcal{R}}\left(\mathbf{X}(0) - \mathbf{X}^{(b)}\right) + \lambda(0) \tag{1.103}$$

For better understanding the impact of the regularization on the estimation, it is interesting to quantify the values of the covariance matrix $B_{\mathcal{R}}$ corresponding to the values of the coefficients α, β and γ in the definition of the cost function J by Equation (1.92) and Equation (1.94).
For doing this, it is first required to invert the matrix $B^{-1}_{\mathcal{R}}$ defined in Equation (1.94). This can not be done on effective applications, due to the large size of the involved state vectors. Moreover, it has no interest apart having a complete knowledge of the links imposed between variables of the state vector and between pixels of the spatial domain. However, this allows visualizing and understanding how the regularization terms act on the estimation results for real applications.

In order to easily compute the inverse of the matrix B^{-1}, we consider a small size image domain Ω of 35×35 pixels. One can extract the $\mathbf{p}^{th}$ line of the covariance matrix that corresponds to the covariance values of the pixel $\mathbf{p} = (i, j)$ with the rest of the domain. As the state vector is composed of the three fields, corresponding to the values of u, v and I at all pixel locations, it is possible to visualize these fields as images. We focus on the visualization of the covariances in $B_{\mathcal{R}_{11}}$. The term regularizing the norm of the motion field, $\mathcal{R}_3$ (with parameter γ), acts only on the variances. Varying the parametrization of the two other terms, $\mathcal{R}_1$ (with parameter α), regularizing the gradient norm of $\mathbf{w}$, and $\mathcal{R}_2$ (with parameter β), regularizing the divergence of $\mathbf{w}$, allows to display the covariances between a reference point and the rest of the domain for the motion component u. This leads to Figure 1.1, which displays, for the component u, the covariance values of pixel $(17, 17)$ with the rest of the domain Ω. On the left, the term regularizing the divergence, $\mathcal{R}_1$, is preponderant. In the middle, $\mathcal{R}_1$ and $\mathcal{R}_2$ are of same importance in the computation. On the right, $\mathcal{R}_2$ is preponderant. It can be seen that $\mathcal{R}_1$ seems to mimic a covariance function,

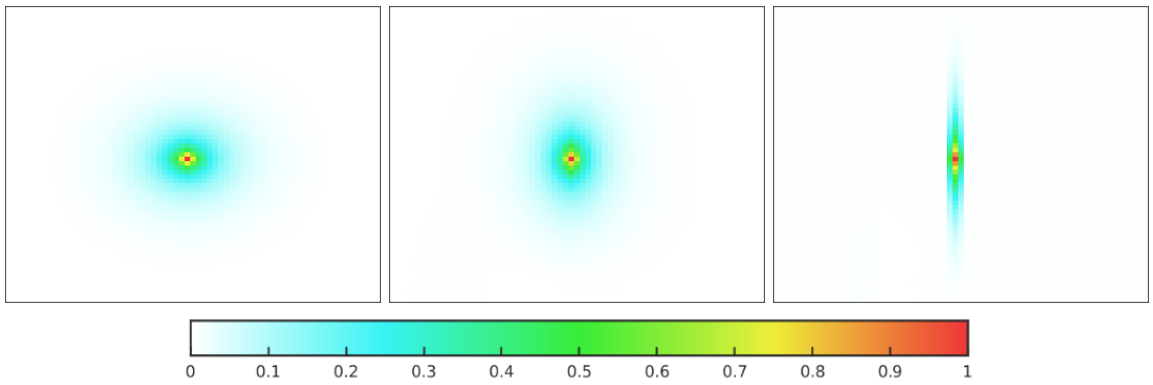

Figure 1.1 – Covariance values associated to the central point *(red pixel)*; when $\mathcal{R}_1$ is preponderant *(on the left)*; when $\mathcal{R}_1$ and $\mathcal{R}_2$ are of same weight *(in the middle)*; and when $\mathcal{R}_2$ is preponderant *(on the right)*.

whose values are issued from an homogeneous diffusion process. In contrary, using $\mathcal{R}_2$ favors some directions.

The range of the covariance values is parametrized by the values of α and β. This is clearly visible on Figure 1.2. This figure displays the covariance values according to a small α, on the left, and a higher one, on the right, only considering the regularization term $\mathcal{R}_1$.

1.5 Minimization

The motion field corresponding to the minimum of the cost function, given in Equation (1.92), is the motion field that: best explains the displacement underlain in the observations, stays close to the given background value, verifies the evolution model, and satisfies the chosen smoothness properties. In order to compute the result, the minimization is done with the L-BFGS algorithm, as described by Zhu et al. in [32] or Morales et al. in [38]. L-BFGS stands for **L**imited memory **BFGS** and denotes an improvement of the so-called **B**royden- **F**letcher-**G**oldfarb-**S**hanno method. As the original BFGS algorithm, L-BFGS is an iterative quasi-Newton method that computes the argument of the minimum of a specified cost function. L-BFGS relies on an approximation of the Hessian matrix of the cost function. The main difference between BFGS and L-BFGS is the

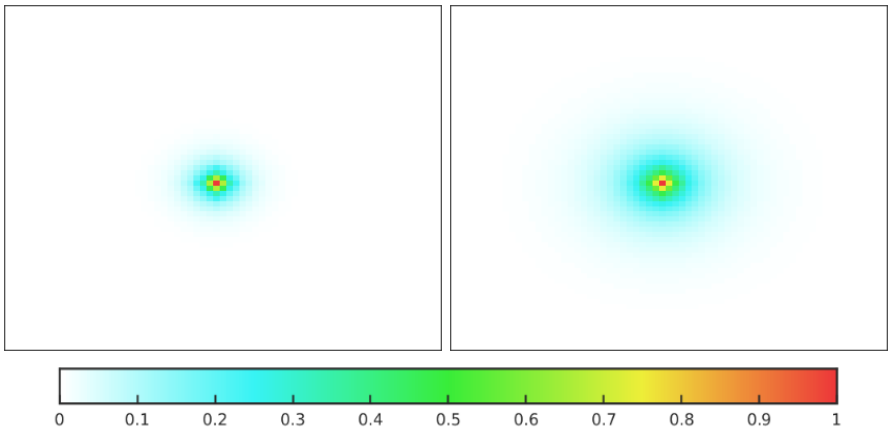

Figure 1.2 – Covariance values associated to the central point *(red pixel)* for $\mathcal{R}_1$ only; with a small α *(on the left)* and a higher one *(on the right)*.

memory required for the computations. With BFGS, the whole Hessian matrix is approximated, while L-BFGS implicitly approximates the Hessian matrix using only a few vectors. The memory needed to achieve the minimization is therefore drastically reduced and renders the method well suited for problems with a large number of variables, i.e. large state vectors.

With L-BFGS, minimization can, as usual, be stopped either with a convergence criterion or with a maximum number of iterations. Figure 1.3 displays the value of the cost function J, at iteration k, divided by the initial value J^0 of the cost function at the first iteration. Each curve corresponds to a different assimilation experiment and represents the decreasing values of the cost function. Colors are only used to better differentiate the curves. The end values of the cost

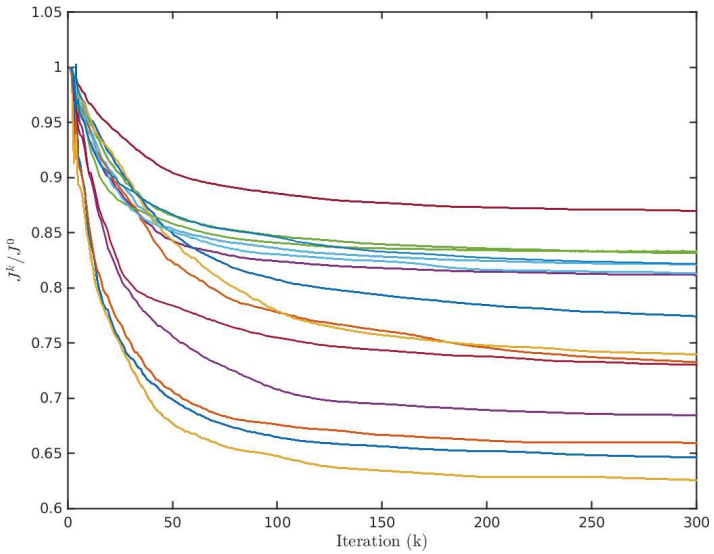

Figure 1.3 – Values of the cost function at iteration k divided by the one at initialization, depending on the number of iterations during the minimization. Each curve corresponds to a different experiment.

function, presented here for 20 different experiments, are depending on the observations, on the background value and on the coefficients of the regularization terms. The interesting features of the figure are the slope of these curves. It is clear, by looking at the slopes, that the minimizer, after two hundreds iterations, has reached a minimum. In fact, as expected, the minimizer, beginning from the background as starting point, first rapidly decreases the cost function value. Then, after fifty iterations, the minimization is slower and, after two hundred iterations, there is nearly no improvement in the minimization. This remark is important as it gives us an upper bound of the computational time needed for the minimization, which is crucial in an operational implementation.

1.6 Pseudo-code

Having defined the cost function $J(\mathbf{X}(0))$ and its gradient $\dfrac{\partial J}{\partial \mathbf{X}(0)}$ and described the method for computing the minimum of the cost function $J(\mathbf{X}(0))$, the whole algorithm of image assimilation may be described by the few lines of pseudo-codes, which are given in Algorithm 1. In that context, the index $\cdot^k$ denotes values at iteration number k of the minimization process.

Algorithm 1 4D-Var algorithm.

1: Read $\mathbf{X}^{(b)}$, $\mathbf{Y}$, B, R_I
2: $k = 0$
3: $\mathbf{X}(0)^k = \mathbf{X}^{(b)}$
4: **while** $|\dfrac{\partial J^k}{\partial \mathbf{X}(0)}| > \mathcal{E}_1$ and $\dfrac{||\mathbf{X}(0)^k - \mathbf{X}(0)^{k+1}||}{||\mathbf{X}(0)^k||} > \mathcal{E}_2$ and $k <$ MaxIter **do**
5: Run model and compute $J(\mathbf{X}(0)^k)$
6: Run adjoint model and compute $\dfrac{\partial J}{\partial \mathbf{X}(0)^k}$
7: $\mathbf{X}(0)^{k+1} = L - BFGS\left(\mathbf{X}^k, J(\mathbf{X}^k), \dfrac{\partial J}{\partial \mathbf{X}^k}\right)$
8: $k = k + 1$
9: **end while**
10: Return $\mathbf{X}(0)^k$

1.7 Conclusion

This chapter described a variational data assimilation method for motion estimation from image sequences. This approach relies on an Image Model $\mathbb{M}$, integrating a state vector $\mathbf{X}$ over time, and some image acquisitions, used as observations $\mathbf{Y}$ in the assimilation process. The definition of a cost function computing the discrepancy between $\mathbf{X}$ and $\mathbf{Y}$ allows to minimize their difference through an iterative approach. The result is the optimal compromise between the model outputs and the observations. However, the use of regularization functions can be mandatory in order to avoid computing physically absurd state functions. The description of such regularization terms is given and some interpretations of the background error covariance matrix are shown. The method described in this chapter, to simplify further discussions, will be called the first method and denoted $\mathcal{M}_1$ in the rest of this manuscript.

Considering a model error would be one of the major improvements of $\mathcal{M}_1$. This consists in adding a third error term, $\mathcal{E}_Q$, in the assimilation system:

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{x},t) + \mathbb{M}(\mathbf{X})(\mathbf{x},t) = \mathcal{E}_Q \tag{1.104}$$

$$\mathbf{X}(\mathbf{x},0) = \mathbf{X}^{(b)}(\mathbf{x}) + \mathcal{E}_B(\mathbf{x}) \tag{1.105}$$

$$\mathbb{H}(\mathbf{X},\mathbf{Y})(\mathbf{x},t) = \mathcal{E}_R(\mathbf{x},t) \tag{1.106}$$

$\mathcal{E}_Q$ would model the error on the temporal evolution and leads to the so-called weak 4D-Var formulation. The interested Reader should consider the article of Courtier et al. [39] or the technical report from Valur [40]. This model error term is used by Béréziat et al. in [41] for assessing geophysical forces and in [42] for computing acceleration.

Chapter 2

Image structures and motion estimation

Starting from the remark that regions of high spatial gradient present a strong and reliable information for estimating motion, this chapter proposes an improvement of Method $\mathcal{M}_1$, where structures are considered in the estimation process. The involvement of a structures representation in the assimilation process is obtained by adding variables describing the structures in the state vector. A first study characterizes the structures displayed on the image sequence by their boundary. This description is discussed in Section 2.1. In that case, the observation vector includes not only the gray level function, but is expanded with a characterization of the structures, defined by the distance map to the contour points computed on the acquisitions. This is described in Section 2.2. The state vector including a new variable, the Image Model $\mathbb{IM}$ is redefined for describing the displacement of structures on the image sequence. These modifications of $\mathbb{IM}$ are given in Subsection 2.3. It is shown, in Section 2.4, that only a few modifications of Method $\mathcal{M}_1$, presented in Chapter 1, will have to be done for being compatible with this new design. Estimation results are analyzed in Chapter 4 and an operational application on precipitations nowcasting will be described in Chapter 5.

2.1 Structures representation

The goal of this section is to define the mathematical representation of the structures displayed on an image sequence. This representation is then included in the state vector and in the assimilation process in order to better the motion estimation and to track the structures on the sequence.

The choice is made to model a structure according to its boundary. If the structure has one connexe component, its boundary $C(t)$, at date t, is defined by:

$$C(t) := \{\mathbf{x}(s,t),\ s \in [0,1] \mid \mathbf{x}(0) = \mathbf{x}(1)\} \tag{2.1}$$

As connexe components can merge or split on the temporal interval corresponding to the studied image sequence, their characterization should verify the same property and allow the topology of a structure to vary over time, with 1, 2 or more components. This is particularly important for modeling structures such as clouds and regions with high precipitation rates on meteorological satellite acquisitions or warm water currents on ocean images. Consequently, the mathematical representation of the whole structure, with one or more components, is chosen as an implicit

function, whose value is null on the boundary $C(t)$ and corresponds to the signed distance to the closest point of $C(t)$ at each other pixel of the image domain. This is defined by Equation (2.2):

$$\phi(t) = \begin{cases} 0 & \text{if } \mathbf{x} \in C(t) \\ -d(\mathbf{x}, C(t)) & \text{if } \mathbf{x} \text{ is outside the structure} \\ d(\mathbf{x}, C(t)) & \text{if } \mathbf{x} \text{ is inside the structure} \end{cases} \tag{2.2}$$

where d is the function computing the Euclidean distance between the point $\mathbf{x}$ and the curve $C(t)$. Therefore, values are positive within the boundary $C(t)$ and negative outside, as illustrated on Figure 2.1. The implicit function $\phi(t)$ will, from now on, be named distance map and it will be used for characterizing the structures in the state vector.

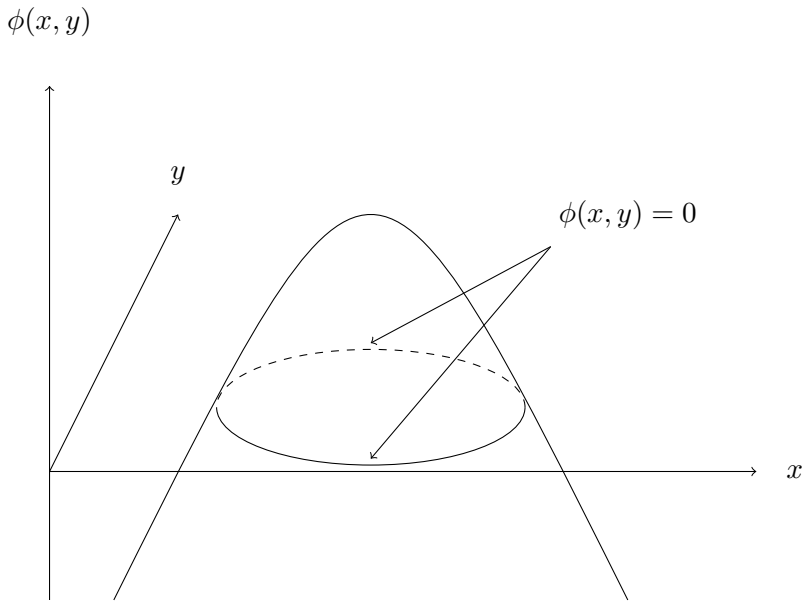

Figure 2.1 – Implicit function used to represent the boundary of the structures.

Reversely, the boundary $C(t)$ can easily be computed from the implicit function ϕ, according to the following equation:

$$C(t) = \{\mathbf{x}(t) | \phi(\mathbf{x}, t) = 0\} \tag{2.3}$$

In the following, going from ϕ to C, and reciprocally, is considered trivial and no more discussed.

The distance map ϕ is added in the state vector $\mathbf{X}$, defined in Chapter 1, which leads to the new state vector (for sake of simplicity, we keep the same notation):

$$\mathbf{X} = \begin{pmatrix} \mathbf{w} \\ I \\ \phi \end{pmatrix} \tag{2.4}$$

that includes the motion field, the image function and the structures representation.

2.2 Definition of observations and of observation operators

In Chapter 1, the image acquisitions I^O, on which motion is estimated, were directly compared to the image function $I(t)$, included in the state vector, at acquisition dates. Their discrepancy

was the quantitative measure used by the image assimilation process. As we have now included a description of the structures in the state vector, it is required to characterize the structures on the image acquisitions I^O, with a variable denoted by ϕ^O. This characterization of the structures on the image acquisitions, ϕ^O, is then added in the observation vector $\mathbf{Y}$, which leads to:

$$\mathbf{Y} = \begin{pmatrix} I^O \\ \phi^O \end{pmatrix} \tag{2.5}$$

It is also required to compare the state vector variable ϕ with its equivalent ϕ^O, in the observation vector $\mathbf{Y}$. The observation operator $\mathbb{H}$ has then to be modified to compute both the discrepancy between images and the difference between structures representation.

The method, which is applied to compute the representation of structures on the image acquisition I^O depends on the application domain. Let discuss, as illustration, the issue of clouds on meteorological satellite images. The boundary of a cloud is hard to precisely locate if there is a lot of filaments on the image. On the contrary, a simple threshold can be used to segment rain cells. According to these remarks, two approaches have been defined and described in the following. In the first case, the structures are obtained by contour points and, in the second, they are defined by thresholding the images.

The first approach uses the contour points computed on the images for characterizing the structures. The main drawback is that the resulting boundaries are now open and it is no more possible to define the inside and outside of a structure. Contour points are obtained by applying the edge detection algorithm defined by R. Deriche in [43]. The method is based on the three criteria defined by J. Canny [44]: a low error rate; an accurate localization; a given contour is only marked once. First, the gradient is obtained by computing, on a smooth version of the image, the derivatives in each direction. Then, the norm of the gradient is computed and local maxima of the gradient norm, in the direction of the gradient vector, are extracted. Last, an hysteresis thresholding is applied on these maxima resulting in the contour image.
Having computed the contour points on an image acquisition, it is then straightforward to define, for each pixel, its distance to the closest contour point. As the boundaries are not closed curves, it is impossible to sign this distance map for coding the belonging of a pixel to one structure. Consequently, the components ϕ and ϕ^O can not be directly compared by computing their difference. The absolute value of ϕ has to be taken before comparison with ϕ^O.

Deriche's filter computes contour points on the acquisition as locations of high contrast. As a result, some of the contour points are not compatible with the structure, as seen on Figure 2.2, and correspond to local contrasts.

The structures representation ϕ, which has been included in the state vector $\mathbf{X}$, tends to describe global features of the image. Its comparison with the contour points computed by the Deriche filter may lead to inappropriate results. To limit the spurious comparisons and the computational time, we consider the fact that the boundary $C(t)$ is supposed to move from a maximum of s pixels during each time step of the temporal integration. From t to $t+1$, the boundary $C(t)$ stays within a region $\Omega_s(t)$, defined from $C(t)$, and the comparison is only required on these pixels. A function ζ_s is then defined on the image domain Ω, such that $\zeta_s(\mathbf{x})$ is equal to 1 near the boundary $C(t)$ and decreases continuously from 1 to $\zeta_s(\mathbf{x}) < 10^{-5}$, when going away from $\Omega_s(t)$.

As a conclusion from the previous discussions, the component of the observation operator, related to the structures representation, is written as:

$$\mathbb{H}_\phi(\mathbf{X}, \mathbf{Y}) = \zeta_s(\mathbf{x}) \left(|\phi| - \phi^O \right) \tag{2.6}$$

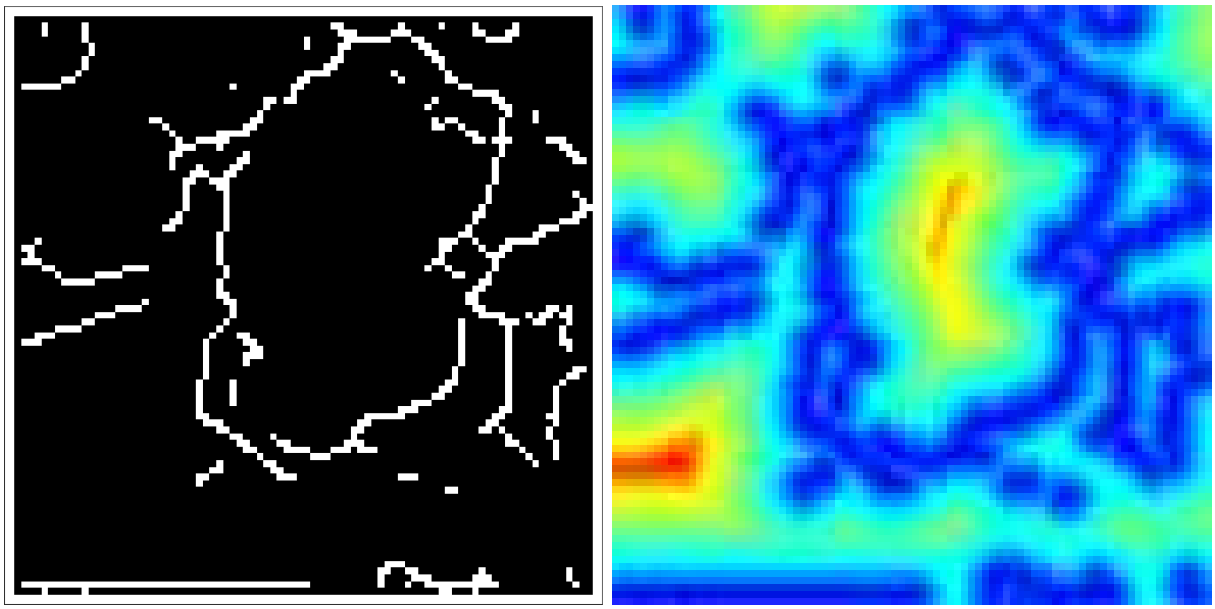

Figure 2.2 – Contour points (*on the left*) and the unsigned distance map to these points (*on the right*).

The second approach for representing the structures consists in applying a threshold to the whole set of observed images. This provides a number of connexe components, with a closed boundary on each acquisition, and allows to compute the associated signed distance functions ϕ^O. In that case, the component of the observation operator related to the structures representation is written as a simple difference between the signed distance map ϕ of the state vector $\mathbf{X}$ and the variable ϕ^O computed on the image acquisition I^O:

$$\mathbb{H}_\phi(\mathbf{X}, \mathbf{Y}) = \phi - \phi^O \tag{2.7}$$

Figure 2.3 displays, on the left, the binary representation of high precipitation regions obtained after applying a given threshold and, on the right, the distance map to these regions.

2.3 Image Model including a structures representation

In order to use the semantic information included in the structures representation for the assimilation process, thanks to the observation operator $\mathbb{H}_\phi(\mathbf{X}, \mathbf{Y})$, the Image Model $\mathbb{M}$ has to be modified. This model should also describe the temporal evolution of the implicit function ϕ, compatible with the motion field $\mathbf{w}$. Let remind that the state vector $\mathbf{X}$ is defined by:

$$\mathbf{X} = \begin{pmatrix} u \\ v \\ I \\ \phi \end{pmatrix} \tag{2.8}$$

The model $\mathbb{M}$ is obtained by the following system of equations describing the evolution in time

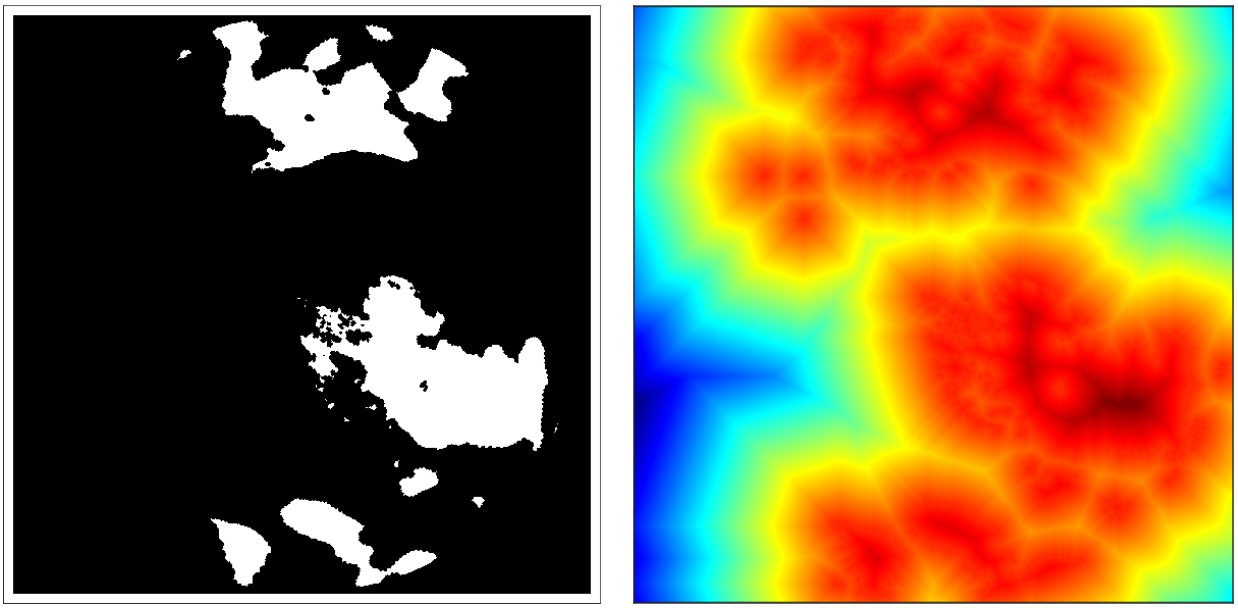

Figure 2.3 – Threshold of the precipitation rate (*on the left*) and the signed distance map to the structures (*on the right*).

of the four variables of the state vector $\mathbf{X}$:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y} = 0 \tag{2.9}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y} = 0 \tag{2.10}$$

$$\frac{\partial I}{\partial t} + \mathbf{w} \cdot \nabla I = 0 \tag{2.11}$$

$$\frac{\partial \phi}{\partial t} + \mathbf{w} \cdot \nabla \phi = 0 \tag{2.12}$$

This is summarized, after temporal discretization, by:

$$\mathbf{X}_{t+1} = \mathbb{M}(\mathbf{X}_t) \tag{2.13}$$

The first three equations on u, v and I are the same than for the Image Model presented in Section 1.2 of Chapter 1. The last one, Equation (2.12), expresses that the distance map ϕ is transported by the motion field, simultaneously with the image function. This comes from the remark that, the boundary of the structure being defined from the gray level function, the distance map is transported accordingly to the image function.

The next section describes the discretization schemes and the difficulties appearing when using an implicit function for characterizing distances.

2.3.1 Discretization and reinitialization module

The discretization of Equations (2.9, 2.10, 2.11) has been discussed in Chapter 1 and is still applied in this new context. The following discussion is then only concerned with the discretization of Equation (2.12).

When integrating in time the model presented corresponding to Equations (2.9, 2.10, 2.11, 2.12), the goal of data assimilation is to compare the state vector $\mathbf{X}$, at the acquisition dates, with the observation vector $\mathbf{Y}$. For performing a pertinent comparison between the distance map ϕ^O, computed on the image acquisition I^O, and the function ϕ included in the state vecteur $\mathbf{X}$, the temporal integration of Equation (2.12) should preserve the mathematical properties of this function. Therefore, $\mathbb{M}$ should preserve, on the variable ϕ, the two following properties:

- ϕ should always represent the distance to the closest boundary point,
- the boundary C, corresponding to ϕ should be a closed curve characterizing the structures.

As discussed in [45], almost no integration scheme allows to preserve the property of being a distance map. Mathematically, ϕ is a distance map if, and only if, the following equation is verified:

$$||\nabla\phi(t)|| = 1 \tag{2.14}$$

Two approaches have been studied to keep both properties, distance map and closed curve, during the temporal integration of the Image Model $\mathbb{M}$ that is applied by the data assimilation method.

The first approach consists in computing a modified motion field $\mathbf{w}_m(t)$ from the motion field $\mathbf{w}(t)$ described by the model. This field $\mathbf{w}_m(t)$ is then used to transport $\phi(t)$ and allows preserving $||\nabla\phi(t)|| = 1$. Such a motion field can be computed, on the whole image domain Ω, by using the Fast Marching method, described by Sethian in [46]. The Fast Marching method, combined with the discretization schemes defined by Sethian, also ensures that $C(t)$ stays a closed curve during the whole temporal integration. Therefore, the Fast Marching method has been strongly considered for the definition of our image assimilation framework. However, the 4D-Var data assimilation technique requires the computation and the backward integration of the adjoint model of $\mathbb{M}$, as described by Le Dimet and Talagrand in their founding paper [26]. The advection of the distance map, as done in the procedure of Sethian, does not update the value of $\phi(t)$ in a predetermined order of pixels of the spatial domain. Computation and application of the gradient of the Fast Marching module would then necessitate to keep in memory, at each integration step, the order in which the pixels are updated. This information is mandatory for the backward integration of the adjoint that has to follow exactly the same order. The memory cost is then highly increased and renders this solution unaffordable for usual applications with images. Therefore, the idea of computing a modified motion field $\mathbf{w}_m(t)$ for the temporal integration has been left over and an alternative approach, based on a reinitialization module, as proposed by Sussman et al. [47], has been considered.

Let now describe the design of the reinitialization module.
At each date $(t-1)$, $\phi(t-1)$ is transported by $\mathbf{w}(t-1)$, using Equation (2.12) of the Image Model $\mathbb{M}$, in order to obtain the value $\phi(t)$. As the result of one integration step is no more a distance map, the function has to be corrected by the reinitialization module to recover the property $||\nabla\phi(t)|| = 1$. Let temporary denote ϕ_t the value of $\phi(t)$ that is obtained from the advection of $\phi(t-1)$. The reinitialization process, introduced in Sussman et al. [48], consists to find the steady state ψ_l (index l stand for *l*imit) of the following partial differential equation:

$$\frac{\partial\psi}{\partial\tau} = \text{Sign}(\phi_t)(1 - ||\nabla\psi||) \tag{2.15}$$

with initial condition: $\psi_0 = \phi_t$. At the limit, the steady state ψ_l is a distance map that verifies $||\nabla\psi_l(\mathbf{x})|| = 1$, for all locations $\mathbf{x}$, according to Equation (2.15). If the process also forces ψ_l to have the same zero values than ϕ_t, then ψ_l is the distance map characterizing $C(t)$. $\phi(t)$ then gets the value ψ_l before further integration from t to $t+1$, and so on.

However, as previously stated, the gradient property of ϕ has only to be valid in the region $\Omega_s(t)$ surrounding $C(t)$, because the curve's displacement should be less than s pixels. Therefore, integration of Equation (2.15) is stopped when the property $||\nabla\psi_l(\mathbf{x})|| = 1$ is valid for $|\psi_l(\mathbf{x})| < s$. This means, according to Sussman et al. [48], that one only needs $\frac{s}{\Delta\tau}$ integration steps of Equation (2.15), $\Delta\tau$ being the time step used for discretizing Equation (2.15). As the number of integration steps is known in advance and denoted N_ψ, the adjoint used by 4D-Var may be efficiently implemented with a reasonable computational cost and memory requirements.

The previously described reinitialization module is then added to the evolution equation of ϕ, Equation (2.12), in the Image Model $\mathbb{M}$. One integration step of the evolution model (2.13) corresponds to the following description in pseudo-code:

Algorithm 2 Transport of the distance map $\phi(t)$, including the reinitialization module.

Read $\mathbf{X}_0$
$t = 0$
while $t \leq T$ **do**
    $\mathbf{X}_t = \mathbb{M}(\mathbf{X}_{t-1})$
    $\psi^0 = \phi_t$
    while $k < N_\psi$ **do**
        Compute $\psi^{k+1} = \psi_t^k + dl \times Sign(\phi_t) \times (1 - ||\nabla\psi^k||)$
    end while
    $\phi_t = \psi^{k+1}$
end while

This reinitialization module ensures that, at any date and whatsoever the spatial discretization scheme used, the advected structure function ϕ verifies the two properties of being a distance map and representing a closed curve. Therefore, any of the two discretization schemes, described in Section 1.2 of Chapter 1 and applied for advecting I, can be used to advect the distance map ϕ, before calling the reinitialization module for corrections. The model being properly defined and its integration allowing the comparison between $\mathbf{X}$ and $\mathbf{Y}$ at acquisition dates, it leaves to describe the assimilation process that is applied to obtain the estimation.

2.4 Estimation

In order to obtain an estimation of the state vector $\mathbf{X}$, based on the structures displacement, simultaneously with the tracking of the considered structures, the system of equations involved in the image assimilation process is the same as the one presented in Chapter 1 for the estimation with Method $\mathcal{M}_1$. It includes the three following equations, respectively named evolution equation,

background equation and observation equation:

$$\frac{\partial \mathbf{X}}{\partial t}(t) + \mathbb{M}(\mathbf{X})(t) = 0 \tag{2.16}$$

$$\mathbf{X}(0) = \mathbf{X}^{(b)} + \mathcal{E}_B \tag{2.17}$$

$$\mathbb{H}(\mathbf{X}, \mathbf{Y})(t) = \mathcal{E}_R(t) \tag{2.18}$$

At that stage, most of the mathematical components of the system have already been described in Chapter 1. They will not be considered again and we will only give references to their previous description. In the following, we will only focus on giving information on the new issues.

The background of $\phi(0)$, denoted $\phi^{(b)}$, has to be defined and used in Equation (2.17). $\phi^{(b)}$ defines, at initial date 0, the structures to be tracked (or the different connexe components of one structure) on the sequence, in order to improve the estimation of motion. $\phi^{(b)}$ is computed on the background $I^{(b)}$ of the image component I. $\phi^{(b)}$ can be obtained by applying a threshold on the image $I^{(b)}$ or by any other method that allows to segment structures and compute a signed distance map to their boundary. If $\mathbb{P}_{I\phi}$ denotes the projection of the state vector $\mathbf{X}$ on its two components I and ϕ, Equation (2.17) is rewritten, in the context of this chapter, as:

$$\begin{pmatrix} I(0) \\ \phi(0) \end{pmatrix} = \begin{pmatrix} I^{(b)} \\ \phi^{(b)} \end{pmatrix} + \mathcal{E}_{B_{I\phi}} \tag{2.19}$$

where $\mathcal{E}_{B_{I\phi}}$ is a zero-mean Gaussian error term describing the discrepancy between the estimation $\mathbf{X}(0)$ and the background $\mathbf{X}^{(b)}$. More exactly, the error $\mathcal{E}_{B_{I\phi}}$ only concerns the discrepancy on the two components I and ϕ and is equal to $\begin{pmatrix} \mathcal{E}_{B_I} \\ \mathcal{E}_{B_\phi} \end{pmatrix}$. The covariance matrix $B_{I\phi}$ associated to this error term is a block diagonal matrix defined by:

$$B_{I\phi} = \begin{pmatrix} B_I & 0 \\ 0 & B_\phi \end{pmatrix} \tag{2.20}$$

B_I and B_ϕ are matrices containing, respectively the a priori error covariances on the image function I and on the distance map ϕ. The acquisition noise is considered as an uncorrelated Gaussian noise, which leads to consider a diagonal matrix B_I. B_ϕ is also taken as diagonal.

The observation equation, described by Equation (2.18), links the observation $\mathbf{Y}$ to the state vector $\mathbf{X}$. The observation operator $\mathbb{H}_{I\phi}$ is composed of two parts, $\mathbb{H}_I$ and $\mathbb{H}_\phi$, which leads to the equation:

$$\mathbb{H}_{I\phi}(\mathbf{X}, \mathbf{Y})(t) = \begin{pmatrix} \mathbb{H}_I(\mathbf{X}(t), \mathbf{Y}(t)) \\ \mathbb{H}_\phi(\mathbf{X}(t), \mathbf{Y}(t)) \end{pmatrix} \tag{2.21}$$

Equation (2.18) is therefore rewritten, according to the context of this chapter, as:

$$\begin{pmatrix} \mathbb{H}_I(\mathbf{X}(t), \mathbf{Y}(t)) \\ \mathbb{H}_\phi(\mathbf{X}(t), \mathbf{Y}(t)) \end{pmatrix} = \begin{pmatrix} \mathcal{E}_{R_I} \\ \mathcal{E}_{R_\phi} \end{pmatrix} = \mathcal{E}_{R_{I\phi}} \tag{2.22}$$

The operators $\mathbb{H}_I$ and $\mathbb{H}_\phi$ have already been described, respectively, in Sections 1.3 and 2.2. The discrepancy between $\mathbf{X}$ and $\mathbf{Y}$ is described by the error $\mathcal{E}_{R_{I\phi}}$, supposed Gaussian and zero-mean, whose covariance matrix is defined by:

$$R_{I\phi} = \begin{pmatrix} R_I & 0 \\ 0 & R_\phi \end{pmatrix} \tag{2.23}$$

R_I and R_ϕ are diagonal matrices containing, respectively, the covariance of the errors $\mathcal{E}_{R_I}$ and $\mathcal{E}_{R_\phi}$. Moreover, $\mathcal{E}_{R_{I\phi}}$ is supposed uncorrelated to $\mathcal{E}_{B_{I\phi}}$.

As previously, the assimilation system, composed of Equations (2.16, 2.19, and 2.22) is solved by minimizing a cost function J. In this context the cost function writes, with respect to the control variable $\mathbf{X}(0)$:

$$\begin{aligned} J(\mathbf{X}(0)) = &\int_\Omega \left(I(0) - I^{(b)}\right) B_I^{-1} \left(I(0) - I^{(b)}\right) \mathrm{d}\mathbf{x} + \int_\Omega \left(\phi(0) - \phi^{(b)}\right) B_\phi^{-1} \left(\phi(0) - \phi^{(b)}\right) \mathrm{d}\mathbf{x} \\ &+ \int_{\Omega_T} (\mathbb{H}_I(\mathbf{X}(t), \mathbf{Y}(t))) R_I^{-1}(t) (\mathbb{H}_I(\mathbf{X}(t), \mathbf{Y}(t))) \mathrm{d}t\mathrm{d}\mathbf{x} \\ &+ \int_{\Omega_T} (\mathbb{H}_\phi(\mathbf{X}(t), \mathbf{Y}(t))) R_\phi^{-1}(t) (\mathbb{H}_\phi(\mathbf{X}(t), \mathbf{Y}(t))) \mathrm{d}t\mathrm{d}\mathbf{x} + \mathcal{R}\left(\mathbf{X}(0)\right) \end{aligned} \tag{2.24}$$

According to the general formulation of the gradient of the cost function J, Equation (1.79), and considering the particular context of this chapter, the gradient of the cost function Equation (2.24) verifies:

$$\frac{dJ}{d\mathbf{X}(0)} = \lambda(0) + \begin{pmatrix} -\alpha\Delta\mathbf{w}(0) - \beta\nabla \operatorname{div}\left(\mathbf{w}(0)\right) + \gamma\mathbf{w}(0) \\ B_I^{-1}\left(I(0) - I^{(b)}\right) \\ B_\phi^{-1}\left(\phi(0) - \phi^{(b)}\right) \end{pmatrix} \tag{2.25}$$

The minimization method still corresponds to what has been described in Section 1.5 of Chapter 1.

The full estimation system has now been described. Taking into account structures in the estimation implies observing and characterizing these structures in $\mathbf{Y}$, expanding the state vector $\mathbf{X}$ to include these structures characterization, modify the model $\mathbb{M}$ in order to model the structures' evolution and, last, design the error covariances and the observation operator $\mathbb{H}$. The next section discusses why going through all these modifications was, in our applicative framework, the only choice affordable to consider structures in the estimation process. It also explains why, by using the position of the structures on the sequence, the estimation is bettered.

2.5 Interpretation

When implementing the method on the discrete spatial domain Ω, the boundary is first described by a curve C, which segments one or more connex components. The distance to C, denoted ϕ, allows to link each point $\mathbf{c}$ of the boundary C with every other grid pixel $\mathbf{p}$ of Ω. This link also expresses the covariance between $\mathbf{c}$ and $\mathbf{p}$. It should be noted that the covariance information could have been included in the background error covariance matrix B. In that case, at each pixel $\mathbf{p}$, the covariance value is chosen according to the accuracy of image acquisition and to the distance of $\mathbf{p}$ to the boundary C.

However, the modification of the background covariance matrix B from a diagonal matrix to a dense matrix including covariance values, is unaffordable for the following reason. If $\mathrm{N}_\mathbf{X}$ denotes the size of the state vector $\mathbf{X}$, the covariance matrix B is of size $(\mathrm{N}_\mathbf{X})^2$. If this matrix is diagonal, the inversion is directly obtained by inverting the diagonal values. If the matrix B is dense, it still has to be inverted for the image assimilation process, as visible in Equation (2.24), describing the function J, and in Equation (2.25), giving its gradient value. This inversion can be done only once and stored for further use. But this would transfer a computational cost issue to a memory problem. For instance, the matrix B requires several hundreds of gigabytes for a 500×500 pixels

image (stored in double precision). Being a huge drawback for a variational assimilation method, the problem of storing the covariance matrix leads to methods in which they are projected into space of lower dimensions, as described for instance by Chabot et al. in [49].

Even if the storage issue is solved, the computation of the cost function, by Equation (2.24), and of its gradient, by Equation (2.25), would lead to dense matrix-vector products during the minimization process, which would increase drastically the number of operations needed at each iteration of the steepest descent method. Therefore, it is preferable to expand the state vector $\mathbf{X}$ rather than to use a dense B covariance matrix.

Understanding the visible impact of ϕ on the results requires a further analysis of its role during the estimation process, which is obtained by the minimization of the cost function $J(\mathbf{X}(0)$.

At each location $\mathbf{p}$ of the discrete image domain Ω, the value of $\phi(\mathbf{p}, 0)$ is the distance to the closest point $\mathbf{c}$, belonging to the boundary $C(0)$ associated with $\phi(0)$. Therefore, $\phi(0)$ relates each pixel $\mathbf{p}$ of Ω to one point $\mathbf{c}$ of $C(0)$. During the image assimilation process, $\phi(0)$ (and therefore $C(0)$) is iteratively modified, from its background value $\phi^{(b)}$, in order that the values $\phi(t)$ correctly describe the boundary C^O of the structures on the image acquisition I^O, at all acquisition dates. Modifying the curve at a given point $\mathbf{c}$ of the boundary $C(0)$ has a strong influence on all pixels $\mathbf{p}$ of the image domain Ω, which have $\mathbf{c}$ as closest point on the boundary. The estimation of the state vector $\mathbf{X}(\mathbf{p})$ and of the motion field $\mathbf{w}(\mathbf{p})$ at all these points is then relying on the estimation at point $\mathbf{c}$. If the point that is located on $\mathbf{c}$ is moved, for being closer to one point $\mathbf{c}^O$ of the observed boundary C^O, this impacts the estimation of the state vector and consequently of motion on all pixels $\mathbf{p}$ that are linked to $\mathbf{c}$. This remark explains why motion is estimated even far away of the structures and of their boundary. An influence value is defined for each point $\mathbf{c}$ of $C(0)$ as:

$$\xi(\mathbf{p}) = \frac{1}{|d(\mathbf{p}, \mathbf{c}) - \phi(\mathbf{p}, 0)|} \tag{2.26}$$

with $d(\mathbf{p}, \mathbf{c})$ being the Euclidean distance between pixel $\mathbf{p}$ and point $\mathbf{c}$. Figure 2.4 gives the influence map related to the point $\mathbf{c}$, displayed by the gray circle, on the boundary $C(0)$, displayed by the black curve.

2.6 Conclusion

This chapter discussed how to include the information, given by the structures visible on an image sequence, for improving the estimation of motion underlain in the sequence.

A first choice was to represent the structures according to their boundary. The signed distance map ϕ to the boundary is then added in the state vector $\mathbf{X}$. Expanding $\mathbf{X}$ with a structures representation implies to modify accordingly the Image Model $\mathbb{M}$ so that it contains the assumptions on the displacement of the structures. Having characterized the position of structures, with ϕ, it is also necessary to compute a similar information ϕ^O on image acquisitions I^O. Two possibilities have been given in order to compute ϕ^O. The first one relies on contours points, which are computed by the Deriche filter. The second one relies on a simple thresholding of the acquisition in order to get a segmentation of the structures. Each strategy is associated with an observation operator $\mathbb{H}$. This motion estimation approach, which represents structures by their boundary, will be called Method $\mathcal{M}_2$ in the following, in order to simplify the discussions, when comparing results between the different approaches.

It should be noted that Method $\mathcal{M}_2$ may be adapted by modifying the structures representation according to the application. An implicit function is well suited for applications where the structures

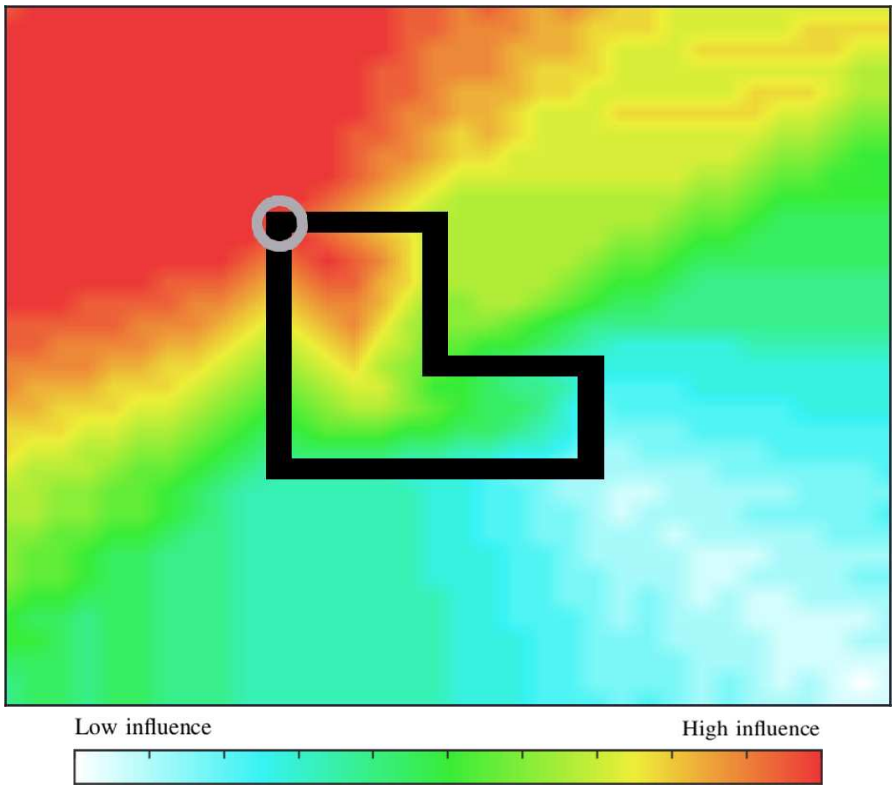

Figure 2.4 – Influence map of the circled point on pixels of the image domain.

topology changes over time. However, a representation of vehicles by quadrangles is more intuitive for traffic monitoring applications. For assessing hurricanes, a representation of the structure by a characteristic point, located on the eye, and a radius value would be sparser. Even if Method $\mathcal{M}_2$ is adaptable according to user requirements, modifying the structures representation implies to redesign the model $\mathbb{M}$, the observation operator $\mathbb{H}$ and the structures observation ϕ^O.

Chapter 3

Representing structures with regularization terms

Chapter 1 described an image assimilation method, named $\mathcal{M}_1$, which was designed for computing motion underlain in an image sequence. The estimation relies on the gray level values of the image acquisitions I^O and on an Image Model $\mathbb{M}$. The mathematical framework that allows to obtain the estimation is called data assimilation. However, estimating motion from an image sequence can be challenging. Therefore, as much information as possible should be considered during the estimation process.

Chapter 2 described a method named $\mathcal{M}_2$, for which the visible structures, displayed on the image sequence, are considered as a new source of information. To include this information, the state vector $\mathbf{X}$ is expanded with a new variable that describes the structures, the distance map ϕ. The Image Model $\mathbb{M}$ is then accordingly modified for being able to describe the temporal evolution of ϕ. Moreover, the observation vector $\mathbf{Y}$ is not only including the image acquisitions I^O but also includes a variable characterizing the structures position, denoted ϕ^O. Section 2.5 of Chapter 2 has stated that Method $\mathcal{M}_2$ correlates each pixel of the discrete domain Ω to the structures boundary. Therefore, another way of characterizing the structures displayed on an image sequence could be done by modeling the background error covariance matrix B. To mimic the effect of structures representation used in Chapter 2, the covariances should be of high value in the direction of the image gradient and have less impact along its normal.

Considering an anisotropic diffusion in order to model the values of the background error covariance matrix B has been discussed in the literature, for instance by Weaver and Ricci in [50]. Their method relies on the existence of a square root of the inverse of the covariance matrix, which is denoted by $B^{-\frac{1}{2}}$. Using this square root matrix, a variable substitution, $\boldsymbol{\zeta} = \left(B^{-\frac{1}{2}}\right)^T \mathbf{X}$, is applied in the cost function J, which is then minimized with regard to $\boldsymbol{\zeta}$. According to Weaver and Ricci, their approach allows every type of anisotropic diffusive regularization as $\left(B^{-\frac{1}{2}}\right)^T$ contains the diffusion parameters. However, the implementation of the 4D-Var algorithm should be modified so that minimization is done according to $\boldsymbol{\zeta}$.

While preserving the implementation of the image assimilation method presented in Chapter 1 it is not affordable to store nor to inverse a dense background error covariance matrix B. However, as discussed in Subsection 1.4.1 of Chapter 1, it is possible to establish a correspondence between a specific type of background error covariance matrix, denoted $B_{\mathcal{R}}$, and the regularization terms.

The main idea is that the covariance values, which are mathematically corresponding to these regularization terms, should be equal to the one resulting from an anisotropic diffusion process. The equivalence between regularization terms and $B_{\mathcal{R}}$ allows: first, to regularize the estimation at a very low computational cost, thanks to the formulation with regularization terms; second, to interpret and visualize the spatial effects on the estimation according to the covariance matrix formulation. Therefore, considerations on the implementation and the computational cost will be discussed with the regularization terms. However, the interpretation and visualization will be done through a discussion on the impact of the matrix $B_{\mathcal{R}}$.

It has been shown, in Section 1.4 of Chapter 1, that the values of the matrix $B_{\mathcal{R}}$ seemed issued from an isotropic diffusion process. However, in order to mimic the effect of the structures representation used by Method $\mathcal{M}_2$, the covariance values should result from an anisotropic diffusion process. Section 3.1 thus describes the design of a novel regularization term $\mathcal{R}_4$, depending on a direction vector d. This direction vector d represents, at point $\mathbf{x}$, the direction to the closest boundary point $\mathbf{c}$ and is defined by:

$$d(\mathbf{x}) = \frac{\mathbf{c}-\mathbf{x}}{||\mathbf{x}-\mathbf{c}||} \tag{3.1}$$

Adding this regularization term $\mathcal{R}_4$ to the cost function makes it possible to model the covariance between pixels of the discrete domain Ω, according to visible structures.

This explanation is followed, in Section 3.2, by a quantification of the computational cost for this new estimation method considering structures and a discussion of its impact on the assimilation process.

Then, a few illustrations of the modeling of the covariance matrix $B_{\mathcal{R}}$ and of its effect on the estimation are given in Section 3.3.

Last, the chapter will end on a few concluding remarks in Section 3.4.

3.1 Directional regularization

This section describes how to design a regularization term, applied on the motion field $\mathbf{w}$, which depends on the direction $d(\mathbf{x})$ at point $\mathbf{x}$, to the closest contour point $\mathbf{c}$. We will also demonstrate that this directional regularization term $\mathcal{R}_4$ allows to model the background error covariance matrix, $B_{\mathcal{R}}$, as if its values were issued from an anisotropic diffusion process. As $\mathcal{R}_4$ will be designed according to the visible structures in the image function I, it will mimic the effect of the structures representation used in Chapter 2.

Before designing $\mathcal{R}_4$, let remind the definition of the regularization terms, $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$, that were used in Chapter 1 in order to ensure the spatial smoothness and the chosen properties of the estimated motion field $\mathbf{w}$:

$$\mathcal{R}_1(\mathbf{X}(0)) = \frac{1}{2}\alpha \int_\Omega ||\nabla\left(\mathbb{P}_{\mathbf{w}}(\mathbf{X}(0))\right)||^2 \mathrm{d}\mathbf{x} \tag{3.2}$$

$$\mathcal{R}_2(\mathbf{X}(0)) = \frac{1}{2}\beta \int_\Omega \left[\mathrm{div}(\mathbb{P}_{\mathbf{w}}(\mathbf{X}(0)))\right]^2 \mathrm{d}\mathbf{x} \tag{3.3}$$

$$\mathcal{R}_3(\mathbf{X}(0)) = \frac{1}{2}\gamma \int_\Omega ||\mathbb{P}_{\mathbf{w}}(\mathbf{X}(0))||^2 \mathrm{d}\mathbf{x} \tag{3.4}$$

For the detailed interpretation, the interested Reader should refer to the discussion given in Section 1.4 of Chapter 1, as only the main issues are summarized below.

The regularization terms are added to the cost function J, which is equal to:

$$J(\mathbf{X}(0)) = \frac{1}{2}\int_{\Omega} (\mathbf{X}(0)-\mathbf{X}_b)^T B^{-1} (\mathbf{X}(0)-\mathbf{X}_b)\,\mathrm{d}\mathbf{x} + \frac{1}{2}\int_{\Omega_T} (\mathbb{H}(\mathbf{X},\mathbf{Y}))^T R^{-1} (\mathbb{H}(\mathbf{X},\mathbf{Y}))\,\mathrm{d}\mathbf{x}\mathrm{d}t + \mathcal{R}_1(\mathbf{X}(0)) + \mathcal{R}_2(\mathbf{X}(0)) + \mathcal{R}_3(\mathbf{X}(0)) \quad (3.5)$$

$\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$ penalize irregular values of $\mathbf{w}$ during the assimilation process. In fact, irregular motion fields $\mathbf{w}$ will get high values of the regularization terms, which automatically increase the value of the cost function J. Consequently, these fields $\mathbf{w}$ will not appear during the minimization process.

In Subsection 1.4.1 of Chapter 1, it has been shown that the following actions are equivalent:

- to simultaneously use the regularization terms of the cost function J and a diagonal background error covariance matrix B, which means considering no covariance between pixels.
- to suppress the regularization terms of the cost function J, to consider that the background error covariance matrix B is non diagonal and give it the suitable covariance values. Let denote $B_{\mathcal{R}}$ this background error covariance matrix containing the values issued from the regularization terms.

The matrix $B_{\mathcal{R}}$, resulting from the second case, depends on the discretization schemes used to compute the regularization terms, $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$, and of their respective weighting coefficients α, β and γ. In Chapter 1, first order upwind schemes are used for the spatial discretization, resulting in a covariance matrix $B_{\mathcal{R}}$, whose values are equal to the ones issued from a diffusion process. The chosen discretization schemes determine how the covariances are modeled, whereas, the values of α, β and γ determine the spatial distance for which covariances will impact the estimation.

In order to include the structures in the assimilation process, the background error covariance matrix $B_{\mathcal{R}}$ should be similar to an anisotropic diffusion matrix. Moreover, the coefficients of the anisotropy characterize the structures. Let further design the regularization term $\mathcal{R}_4$ relying on characteristic features of the structures.

Let first note that the regularization term $\mathcal{R}_1$, defined by Equation (3.2) on the continuous image domain Ω, is a function of the norm of the gradient of the motion field. It can be written:

$$\mathcal{R}_1(\mathbf{w}) = \frac{1}{2}\int_{\Omega} \alpha \left(\partial_x u(\mathbf{x})^2 + \partial_y u(\mathbf{x})^2 + \partial_x v(\mathbf{x})^2 + \partial_y v(\mathbf{x})^2\right)\mathrm{d}\mathbf{x} \quad (3.6)$$

It is clear that, in Equation (3.6), both components of the gradient, along x and y, have the same weight α, in the computation of the term $\mathcal{R}_1$. Therefore, at any point $\mathbf{x}$ of the continuous domain Ω, the directions along x and y are equivalently contributing to the computation of the regularization term, leading to an homogeneous in space covariance. The weight value α, given to $\mathcal{R}_1$, controls the radius of the covariance, as shown on Figure 3.1.

Inspired by the remark that the directions ∂_x and ∂_y are both contributing and equally weighted in $\mathcal{R}_1$, let design $\mathcal{R}_4$. It is based on a penalization term, for regularizing motion according to a direction vector $d(\mathbf{x})$, and is weighted by a coefficient $\xi(\mathbf{x})$, both depending on the considered point $\mathbf{x}$ of Ω.

Before defining $\mathcal{R}_4$, let first remind that the state vector variable characterizing structures, the distance map ϕ used in Chapter 2, can be interpreted as a link between each point $\mathbf{x}$ of the

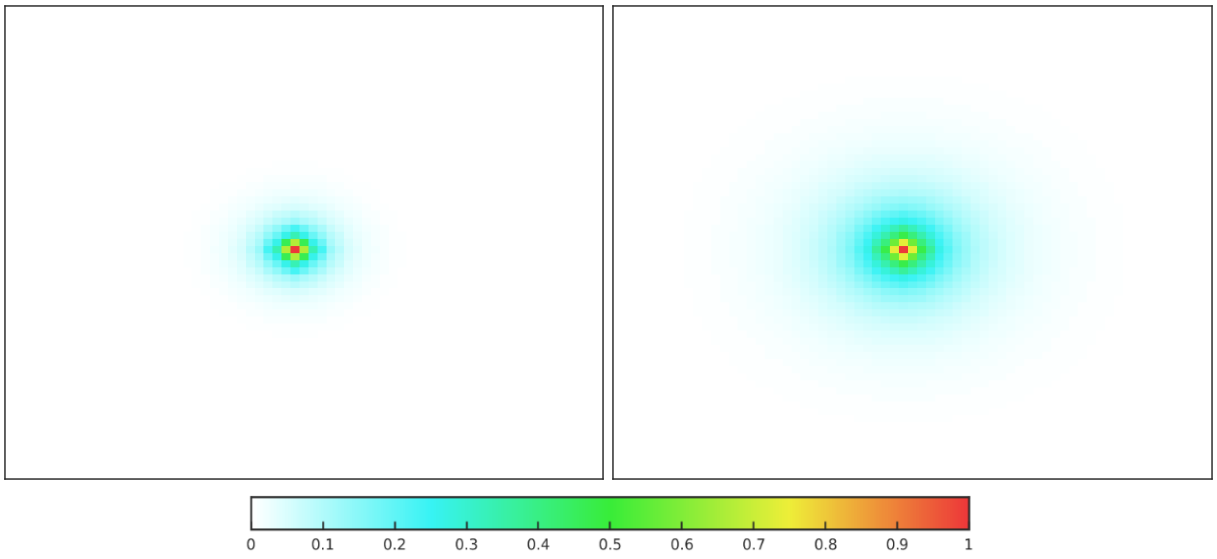

Figure 3.1 – Covariance values according to the central point (*red pixel)* with a small α *(on the left)* and a higher one *(on the right)*.

continuous image domain Ω and the closest point on the structures boundary. For a more complete interpretation of the significance of ϕ, the Reader should go back to Section 2.5 of Chapter 2. The direction vector $d(\mathbf{x})$ involved in $\mathcal{R}_4$ is also linking the location $\mathbf{x}$ to the boundary of structures in order to mimic the impact of ϕ in the assimilation process. As $d(\mathbf{x})$ is the normalized vector of the direction from $\mathbf{x}$ to the closest boundary point $\mathbf{c}$, let define $\partial_{d(\mathbf{x})}$ the derivative along the direction $d(\mathbf{x})$. $\mathcal{R}_4$ is then defined as:

$$\mathcal{R}_4(\mathbf{X}(0)) = \frac{1}{2}\int_\Omega \left(\partial_{d(\mathbf{x})} u(\mathbf{x},0)^2 + \partial_{d(\mathbf{x})} v(\mathbf{x},0)^2\right) \mathrm{d}\mathbf{x} \tag{3.7}$$

Therefore, $\mathcal{R}_4(\mathbf{X}(0))$ computes the norm of the gradient of motion in direction $d(\mathbf{x})$, leading to a regularization along $d(\mathbf{x})$ of the estimated motion field.

In a continuous image domain Ω, $d(\mathbf{x})$ describes an infinite number of directions. After discretization, only 8 directions d_i where $(i = 1, \ldots, 8)$ are possible at pixel $\mathbf{p}$, corresponding to the horizontal, vertical and diagonal neighboring pixels. Let give the values of the discrete filters $\mathcal{D}_{d(\mathbf{p})}$ associated to $d(\mathbf{p})$, with $d(\mathbf{p}) = d_1 \ldots d_8$:

$$\mathcal{D}_{d(\mathbf{p})} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \dfrac{-\xi(\mathbf{p})}{dx} & \dfrac{\xi(\mathbf{p})}{dx} \\ 0 & 0 & 0 \end{bmatrix} \quad \text{if} \quad d(\mathbf{p}) = d_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{or} \quad d(\mathbf{p}) = d_5 = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \tag{3.8}$$

$$\mathcal{D}_{d(\mathbf{p})} = \begin{bmatrix} 0 & \dfrac{\xi(\mathbf{p})}{dy} & 0 \\ 0 & \dfrac{-\xi(\mathbf{p})}{dy} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{if} \quad d(\mathbf{p}) = d_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{or} \quad d(\mathbf{p}) = d_7 = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \tag{3.9}$$

$$\mathcal{D}_{d(\mathbf{p})} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \dfrac{-\xi(\mathbf{p})}{\sqrt{dx^2+dy^2}} & 0 \\ 0 & 0 & \dfrac{\xi(\mathbf{p})}{\sqrt{dx^2+dy^2}} \end{bmatrix} \quad \text{or } d(\mathbf{p}) = d_4 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \text{ if } \quad d(\mathbf{p}) = d_8 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{3.10}$$

$$\mathcal{D}_{d(\mathbf{p})} = \begin{bmatrix} 0 & 0 & \dfrac{\xi(\mathbf{p})}{\sqrt{dx^2+dy^2}} \\ 0 & \dfrac{-\xi(\mathbf{p})}{\sqrt{dx^2+dy^2}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{if } \quad d(\mathbf{p}) = d_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ or } \quad d(\mathbf{p}) = d_6 = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \tag{3.11}$$

where $\xi(\mathbf{p})$ is the weighting parameter, defined at pixel $\mathbf{p}$.

Let recall that each filter $\mathcal{D}_{d(\mathbf{p})}$ can be associated with a row vector $D_{d(\mathbf{p})}$, such that $D_{d(\mathbf{p})}u$ computes an approximation of $\partial_{d(\mathbf{p})}u$ at pixel $\mathbf{p}$. The associated row vector $D_{d(\mathbf{p})}$ contains only 0 values, except at $\mathbf{p}$ and $\mathbf{p}+d(\mathbf{p})$, such that:

$$D_{d(\mathbf{p})}u = u(\mathbf{p}+d(\mathbf{p})) - u(\mathbf{p}) \tag{3.12}$$

Let then define the matrix D_d, whose $\mathbf{p}^{th}$ row is equal to $D_{d(\mathbf{p})}$. Let rewrite the variational formulation of $\mathcal{R}_4$, Equation (3.7), into its discrete formulation with scalar products on the discrete domain Ω:

$$\mathcal{R}_4 = \frac{1}{2}\left(\langle D_d u \ , \ D_d u\rangle + \langle D_d v \ , \ D_d v\rangle\right) \tag{3.13}$$

which is equivalent to:

$$\mathcal{R}_4 = \frac{1}{2}\left(\left\langle u \ , \ D_d^T D_d u\right\rangle + \left\langle v \ , \ D_d^T D_d v\right\rangle\right) \tag{3.14}$$

None of the filters given in Equations (3.8) to (3.11) is used with the formulation (3.14): it is the product of the filter D_d by its transpose that is required during the estimation process.

This regularization term $\mathcal{R}_4$, modeling the structures, is added to the equation of the cost function J:

$$\begin{aligned} J(\mathbf{X}(0)) = \frac{1}{2}\int_{\Omega}(\mathbf{X}(0)-\mathbf{X}_b)^T B^{-1}(\mathbf{X}(0)-\mathbf{X}_b)\mathrm{d}\mathbf{x} \\ + \frac{1}{2}\int_{\Omega_T}\left(\mathbb{H}(\mathbf{X},\mathbf{Y})\right)^T R^{-1}\left(\mathbb{H}(\mathbf{X},\mathbf{Y})\right)\mathrm{d}\mathbf{x}\mathrm{d}t \\ + \mathcal{R}_1(\mathbf{X}(0)) + \mathcal{R}_2(\mathbf{X}(0)) + \mathcal{R}_3(\mathbf{X}(0)) + \mathcal{R}_4(\mathbf{X}(0)) \end{aligned} \tag{3.15}$$

Following the same spirit as in Chapter 1, it is possible to consider the four regularization terms $\mathcal{R}_1$ to $\mathcal{R}_4$ as being part of the inverse of a background error covariance matrix, $B_{\mathcal{R}}^{-1}$, which leads to the new formulation of the cost function:

$$J(\mathbf{X}(0)) = \frac{1}{2}\int_{\Omega}(\mathbf{X}(0)-\mathbf{X}_b)^T B_{\mathcal{R}}^{-1}(\mathbf{X}(0)-\mathbf{X}_b)\mathrm{d}\mathbf{x} + \frac{1}{2}\int_{\Omega_T}\left(\mathbb{H}_I(\mathbf{X},\mathbf{Y})\right)^T R_I^{-1}\left(\mathbb{H}_I(\mathbf{X},\mathbf{Y})\right)\mathrm{d}\mathbf{x}\mathrm{d}t \tag{3.16}$$

As $B_{\mathcal{R}}^{-1}$ replaces the four regularization terms $\mathcal{R}_1$ to $\mathcal{R}_4$, it includes the structures model. According to the computations made in Section 1.4.1 of Chapter 1, $B_{\mathcal{R}}^{-1}$ is equal to:

$$B_{\mathcal{R}}^{-1} = \begin{pmatrix} (\alpha+\beta)D_x^T D_x + \alpha D_y^T D_y + \gamma Id + D_d^T D_d & \beta D_x^T D_y & 0 \\ \beta D_y^T D_x & \alpha D_x^T D_x + (\alpha+\beta) D_y^T D_y + \gamma Id + D_d^T D_d & 0 \\ 0 & 0 & B_I^{-1} \end{pmatrix} \tag{3.17}$$

The motion estimation method, which is based on the regularization term $\mathcal{R}_4$, will be denoted Method $\mathcal{M}_3$ in the following. The next section aims to compare the computational costs of the initial estimation Method $\mathcal{M}_1$, the approach considering the structures through their distance map, Method $\mathcal{M}_2$, and the new Method $\mathcal{M}_3$.

3.2 Computational cost

The additional computational cost for going from the estimation Method $\mathcal{M}_1$ of Chapter 1 to Method $\mathcal{M}_3$, which uses the regularization term $\mathcal{R}_4$ is negligible.

Let, first, consider the increase of the memory required to apply Method $\mathcal{M}_3$ compared to the one used for the application of Method $\mathcal{M}_1$. The filters issued from B^{-1} defined in Method $\mathcal{M}_1$ were the same for each pixel $\mathbf{p}$ of the discrete domain Ω. Whereas, in Method $\mathcal{M}_3$, the filters are depending on the location of the pixel, as seen in the construction of the matrix D_d, from the line vectors $D_{d(\mathbf{p})}$, in Section 3.1. Therefore, the increased need in memory corresponds to the storage of a filter for each pixel $\mathbf{p}$ of the background image. If the coefficients of the filter are stored in double precision, applying Method $\mathcal{M}_3$ for a 500×500 pixels image would result in a 15 megabytes increase of the memory, which is negligible.

Let, now consider the issue of computational cost. As the filters used by Methods $\mathcal{M}_1$ and $\mathcal{M}_3$ have the same size (ie. a matrix three by three), the computational cost of these two methods is strictly identical.

Let remind that Method $\mathcal{M}_2$ requires a modification of the Image Model $\mathbb{M}$ and therefore implies to compute a new adjoint. The integration of the model and its adjoint is the most time consuming part of each image assimilation method, $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$.

Considering structures in the estimation process by Method $\mathcal{M}_3$, rather than by Method $\mathcal{M}_2$, avoids to use an additional variable in the state vector $\mathbf{X}$ and consequently allows to limit the computational time. Moreover, the minimization is done on less variables, which further limits the computational time of the optimization algorithm. Method $\mathcal{M}_3$ is therefore leading to a reduced computational cost in both memory requirements and computing time compared to $\mathcal{M}_2$ and equivalently involves structures in the estimation process.

3.3 Illustrations of the weighting of $\mathcal{R}_4$

Section 3.1 described a new regularization term $\mathcal{R}_4$, weighted by the parameter $\xi(\mathbf{p})$, which relies on the direction vector $d(\mathbf{p})$ from the location $\mathbf{p}$ to the closest boundary point $\mathbf{c}$ on the boundary of the structures. This section will explain how to determine the value of the weight $\xi(\mathbf{p})$ in order to mimic, as close as possible, the impact of the structures on the results obtained by Method $\mathcal{M}_2$.

It has been shown that the regularization terms $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$ and $\mathcal{R}_4$ correspond to the inverse of a background error covariance matrix, $B_{\mathcal{R}}^{-1}$, with non diagonal terms. Similarly to what has

been done in Subsection 1.4.1 of Chapter 1, $B_{\mathcal{R}}$ is computed and analyzed, for improving the understanding and illustrating the ideas. The $\mathbf{p}^{th}$ row of this matrix contains the values of the covariances of a given pixel $\mathbf{p}$ with all other pixels of the discrete domain Ω. As explained in Section 1.1 of Chapter 1, it is straightforward to visualize this row of $B_{\mathcal{R}}$ as an image, displaying the covariance values on the whole image domain Ω.

Let consider having chosen a mathematical representation of the structures displayed on an image. For a description of the processes computing various types of representations, the Reader should go back to Section 2.1 of Chapter 2. From this description of the boundary of structures, it is possible, at each pixel $\mathbf{p}$ of the discrete domain Ω to compute, using Equation (3.1), the direction vector $d(\mathbf{p})$ from $\mathbf{p}$ to the closest boundary point $\mathbf{c}$. Having defined the direction vector $d(\mathbf{p})$, it remains to determine the value of $\xi(\mathbf{p})$, at each location $\mathbf{p}$, for being able to compute the covariances. For visualization purposes, the following analysis relies on the values given to ξ compared to the one given to α. The Reader interested on the impact on the covariances of the parameters β and γ should refer to Subsection 1.4.1 of Chapter 1.

Let consider the case where α and ξ have values of the same order of magnitude and ξ is a constant, not depending of pixel $\mathbf{p}$. This leads to the illustration of the values of $B_{\mathcal{R}}$ given on Figure 3.2 for several pixels of the discrete domain Ω. The figure displays, on the top line, the values of the covariances computed between the red pixel and the rest of the domain. The structure used to determine the direction vectors $d(\mathbf{p})$, for the computation of $\mathcal{R}_4$, is displayed in black. For this illustration, values of $\xi(\mathbf{p})$ are constant over the whole image domain Ω. The bottom line shows the values of the covariances, obtained when using only $\mathcal{R}_1$ with the same value of α. It can be seen that the regularization term $\mathcal{R}_4$ is distorting the initial covariance in the direction of the closest boundary point.

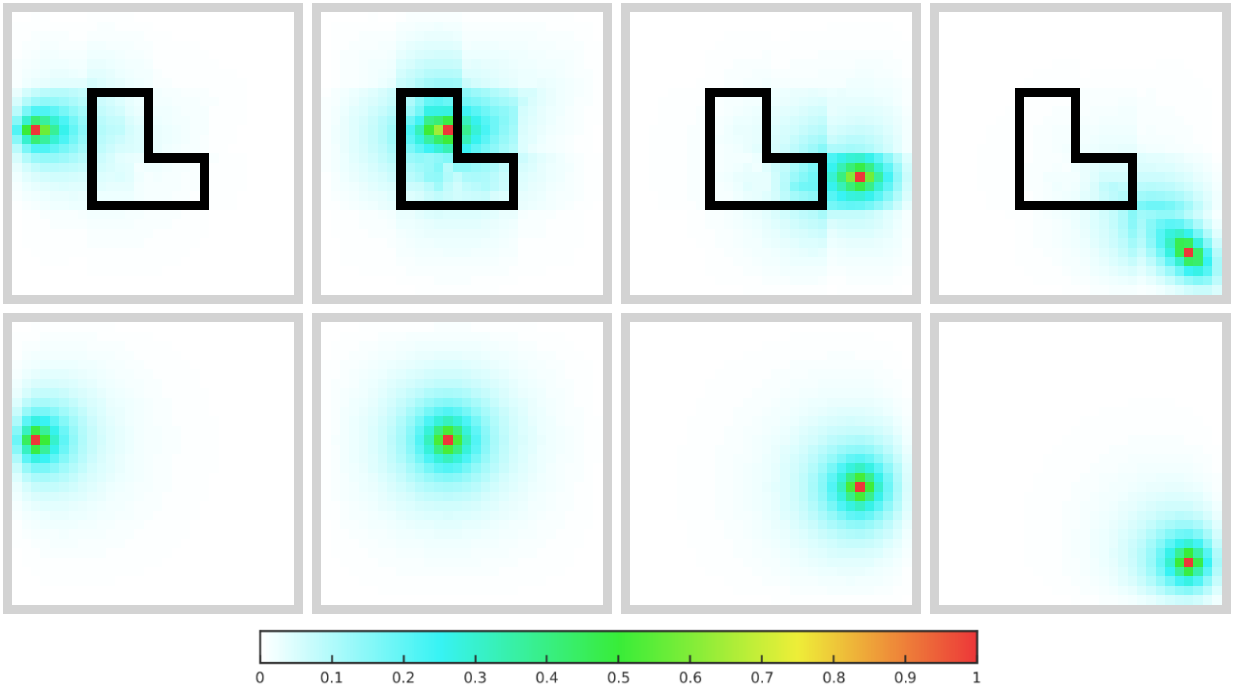

Figure 3.2 – Covariance values computed between the red pixel and the rest of the domain with both $\mathcal{R}_1$ and $\mathcal{R}_4$ *(top line)*, with only $\mathcal{R}_1$ *(bottom line)*. The black curve is used to determine $d(\mathbf{p})$.

It is positive to include the structures in the image assimilation process because the pixels, with a high gradient on the background image provide a reliable information for estimating motion. The result on pixels close to the boundary should therefore be more reliable than those on the pixels far away from the structure. Modeling of the covariances therefore takes the distance to the boundary into account so that pixels far away from the boundary are correlated with those close to the boundary in the estimation.

Let therefore consider the weighting parameter $\xi(\mathbf{p})$ as function of the distance between $\mathbf{p}$ and its closest boundary point $\mathbf{c}$. Figure 3.3 displays the values of the covariance function according to the red pixel and the rest of the domain. It can be seen that the covariance values are higher toward the closest boundary point.

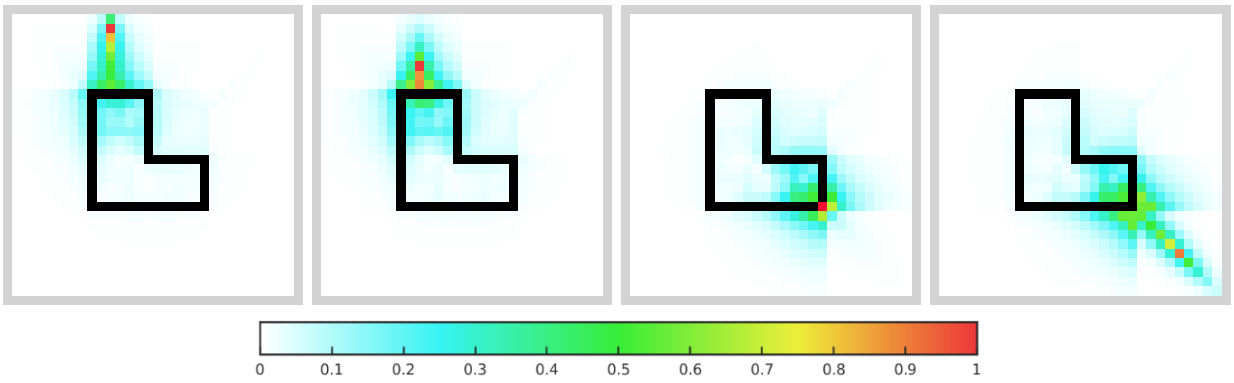

Figure 3.3 – Covariances computed between the red pixel and the rest of the domain. The black curve represents the structure.

If increasing the value of $\xi(\mathbf{p})$, always depending on the distance to the boundary, in order to make $\mathcal{R}_1$ negligible, covariances are shaped into straight lines toward the closest boundary point $\mathbf{c}$, as shown on Figure 3.4.

Let consider $\mathbf{c}$ a point on the structure boundary C. As from Equation 3.1, $d(\mathbf{c})$ is not defined, the choice had been made to consider:

$$\mathcal{R}_4(\mathbf{X}(\mathbf{c},0)) = \mathcal{R}_1\left(\mathbf{X}(\mathbf{c},0)\right) \tag{3.18}$$

As stated, $\mathcal{R}_1$ leads to covariance values issued from a diffusion process, therefore the boundary pixels are considered as diffusive.

Image sequences display motion for different types of scene, for instance clouds moving in the sky or pedestrians moving in front of a wall. In the first case, motion is due to a wind field defined on the whole image domain Ω. Every couple of pixels in the discrete image domain Ω should therefore be more or less correlated during the estimation. In the second case, pedestrians are moving independently one to each other and independently from the background. This independence has to be modeled and covariances between points on different pedestrians should be null. In order to estimate motion of a structure moving independently from the background, the filter $\mathcal{R}_4(\mathbf{X}(\mathbf{c},0))$, at a boundary points $\mathbf{c}$, is modified according to:

$$\mathcal{R}_4(\mathbf{X}(\mathbf{c},0)) = \mathcal{R}_3\left(\mathbf{X}(\mathbf{c},0)\right) \tag{3.19}$$

This regularization term $\mathcal{R}_4(\mathbf{X}(\mathbf{c},0))$ only acts on the given pixel $\mathbf{c}$, without adding any covariance with the surrounding pixels.

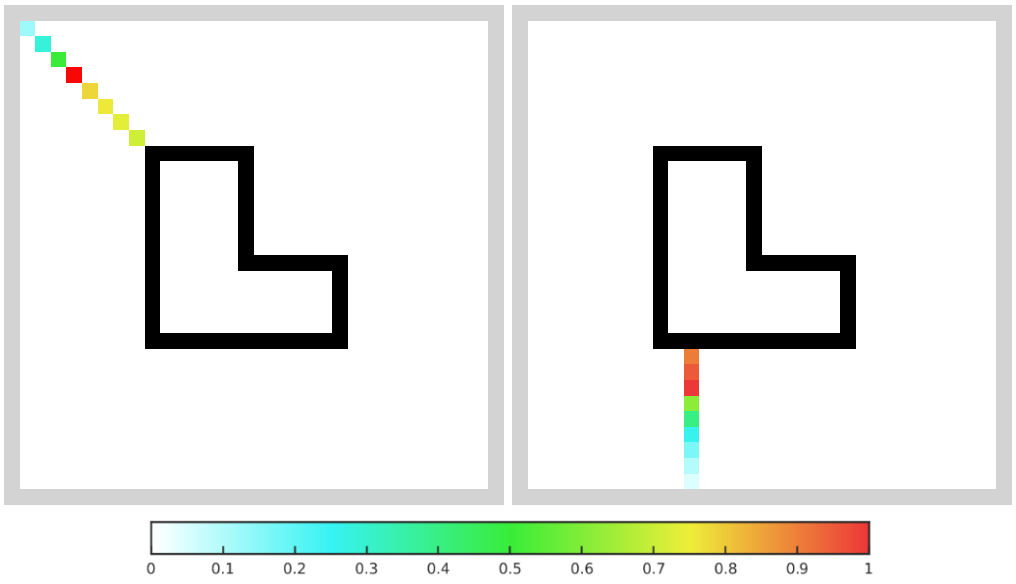

Figure 3.4 – Covariances computed between the red pixel and the rest of the domain by using only $\mathcal{R}_4$. The black curve represents the boundary of the structure.

An illustration of such covariance value, where the structure boundary acts as a correlation limit, is given on Figure 3.5. Pixels inside the structure are uncorrelated to those outside of it. During the estimation, the motion of the structure is estimated independently from the motion outside of it.

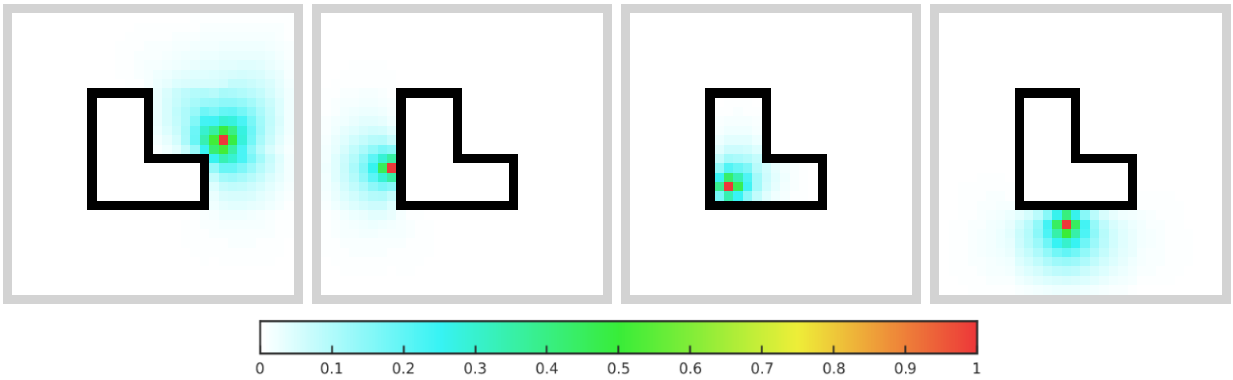

Figure 3.5 – Covariances computed between the red pixel and the rest of the domain. The black curve represents the structure and acts as a limit for the covariance values.

3.4 Conclusion

This chapter described a regularization term named $\mathcal{R}_4$, and its equivalent covariance matrix $B_{\mathcal{R}}$, which models the structures visible on the image sequence.

Involving structures in the image assimilation by modeling the covariances in the matrix $B_{\mathcal{R}}$

with the regularization term $\mathcal{R}_4$ adds almost no cost compared to the initial computational cost of Method $\mathcal{M}_1$.

Method $\mathcal{M}_3$ offers a wide broad of possibilities to design the matrix $B_{\mathcal{R}}$, by modifying the functions computing the direction vector $d(\mathbf{p})$ or the weighting coefficient $\xi(\mathbf{p})$ and some of these possibilities have been illustrated.

Let remark that different types of covariance are included in the matrix $B_{\mathcal{R}}$.

- The matrix is composed of the covariances between the locations $\mathbf{x}_i$ and $\mathbf{x}_j$. The regularization terms $\mathcal{R}_1$ to $\mathcal{R}_4$ model the covariance between values $u(\mathbf{x}_i, 0)$ and $u(\mathbf{x}_j, 0)$ of the motion variable u, as well as between values $v(\mathbf{x}_i, 0)$ and $v(\mathbf{x}_j, 0)$ of the motion variable v. However, the covariances between the values $I(\mathbf{x}_i, 0)$ and $I(\mathbf{x}_j, 0)$ of the image function I have not been modeled and should be investigated in the future.

- The background covariance matrix $B_{\mathcal{R}}$ is also composed of covariances between the variables of the state vector $\mathbf{X}$. A second perspective would be to model $B_{\mathcal{R}}$ such that covariances between the motion $\mathbf{w}$ and the image function I are considered.

Moreover, based on the idea of using regularization terms to model covariances, it should be possible to model covariances in the covariance matrix R of the observation error.

Chapter 4

Experiments and validation

Chapter 1 described a method for estimating motion from an image sequence, based on an evolution model of the studied system. The approach relied on a 4D-Var data assimilation technique.
Chapter 2 improved that initial method by considering the structures displayed on images and their displacement on the sequence, during the motion estimation process. Chapter 2 described the adjustments required by the image assimilation method for being able to include this information on structures while estimating motion. These adjustments consist in expanding the state vector $\mathbf{X}$ with a variable representing the structures, defining the evolution model of that new state variable and an appropriate observation operator, which links the state vector with the observations. These observations are simultaneously the image acquisitions and displayed structures.
All mathematical details having been described in these two chapters, we will now focus on experimental results and discussions about these methods.

Section 4.1 gives results of motion estimation that are based on twin experiments. In such context, the Image Model $\mathbb{M}$ is first integrated, from an initial condition $\mathbf{X}_{GT}(0)$, which is called **G**round **T**ruth, in order to generate an image sequence. Snapshots are extracted from this sequence in order to simulate real satellite acquisition conditions. These snapshots are then used as observations for the assimilation process. Afterward, it is possible to compare, qualitatively and quantitatively, the estimation with the ground truth and to assess the ability of each method to estimate motion. These twin experiments are clearly unrealistic estimation experiments. Their purpose is purely illustrative. They are meant to validate codes and to highlight particularities of each method.

The methods are then applied on real acquisitions in Section 4.2. We consider satellite meteorological images and ground radar acquisitions. Quality of the estimation is hard to assess on this type of data, but some objective criteria allow to draw positive conclusions.

In case of twin experiments, as a ground truth is available, results of estimation are analyzed thanks to statistics that will be given. On real data, as no quantitative result is at hand, the estimated motion fields will be displayed as images. The color coding of the Middlebury database, see [13], will then often be used. It allows to better understand spatial properties of motion as it is often more readable than an arrows representation. Consequently, this color coding allows a comparative assessment of the results obtained by different methods. Figure 4.1 displays the color coding, used in the document. Each direction of motion is represented by a specific hue and the norm is described by the intensity.

Figure 4.1 – Color code for motion display.

4.1 Twin experiments

This section illustrates, through twin experiments, the specificity of each one of the estimation methods described in Chapters 1 and 2 which are respectively named Method $\mathcal{M}_1$ and Method $\mathcal{M}_2$.

Function values, for instance $\mathbf{X}(t_i)$, estimated with a Method $\mathcal{M}$ at date t_i will be denoted $\mathbf{X}_{\mathcal{M}}(t_i)$. As there is no model error involved in our data assimilation systems, the whole estimation only relies on the value of $\mathbf{X}_{\mathcal{M}}(0)$. Comparisons could be done at any date for illustration purposes, but are sufficient at the initial one.

Let recall that, in Chapter 1, the state vector $\mathbf{X}$ of Method $\mathcal{M}_1$ includes the motion field $\mathbf{w}$ and the image component I. The Image Model $\mathbb{M}$ is defined by:

$$\frac{\partial \mathbf{w}}{\partial t} = -\mathbf{w} \cdot \nabla \mathbf{w} \tag{4.1}$$

$$\frac{\partial I}{\partial t} = -\mathbf{w} \cdot \nabla I \tag{4.2}$$

This Image Model is used to generate the image snapshots $I^O(t_i)$ from the initial condition $\mathbf{X}_{GT}(0)$. Motion estimation is then obtained with Method $\mathcal{M}_1$ by assimilating $I^O(t_i)$. The resulting motion field $\mathbf{w}_{\mathcal{M}_1}(0)$ will be compared with the motion component of $\mathbf{X}_{GT}(0)$.

In Chapter 2, when modeling the structures by adding a distance function ϕ in $\mathbf{X}$, the Image Model $\mathbb{M}$ is expanded with the evolution equation of this function:

$$\frac{\partial \phi}{\partial t} = -\mathbf{w} \cdot \nabla \phi \tag{4.3}$$

The first two equations on $\mathbf{w}$ and I remain the same. It should be noted that Equation (4.3) is not used for generating the observed structures positions $\phi^O(t_i)$ in the twin experiment. These structures positions $\phi^O(t_i)$ are directly computed on the image snapshots $I^O(t_i)$ (the one that have been obtained with the first model). Motion estimation is then done with Method $\mathcal{M}_2$, by assimilating the images $I^O(t_i)$ and the distance maps $\phi^O(t_i)$. The resulting motion field, denoted $\mathbf{w}_{\mathcal{M}_2}(0)$, is compared with both motion fields $\mathbf{w}_{GT}(0)$ and $\mathbf{w}_{\mathcal{M}_1}(0)$.

4.1.1 Images without texture

In order to show the interest of using the structures representation ϕ in the state vector $\mathbf{X}$, a first twin experiment is created in which the initial image field displays a small white structure over a dark background. Figure 4.2 displays some of the snapshots $I^O(t_i)$. It should be noted that

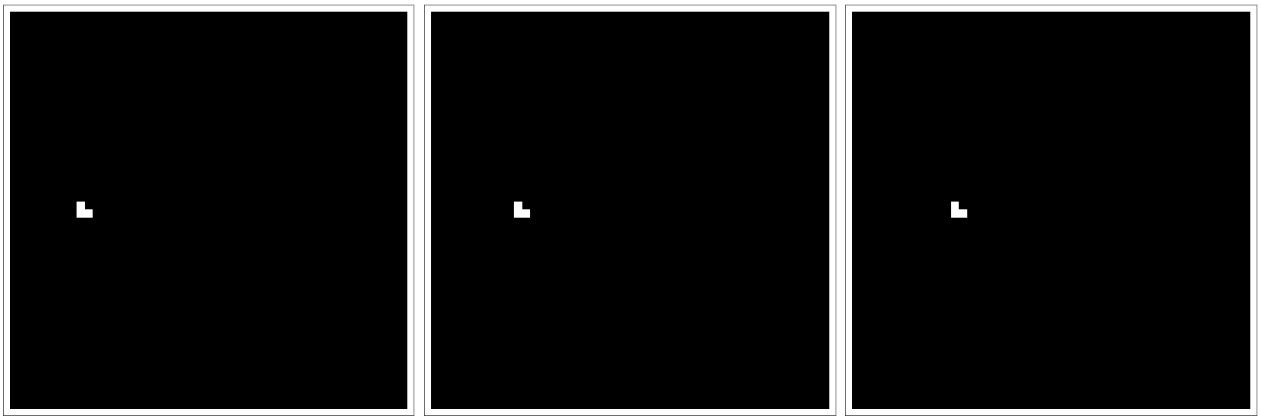

Figure 4.2 – Synthetic sequence of a white structure moving to the right.

there is no texture nor noise on the image data $I^O(t_i)$. This absence of texture results in a lack of information about the overall motion on the image domain. Consequently, Method $\mathcal{M}_1$, which does not include the distance map ϕ, fails to estimate any motion in a large part of the domain, as seen on Figure 4.3.

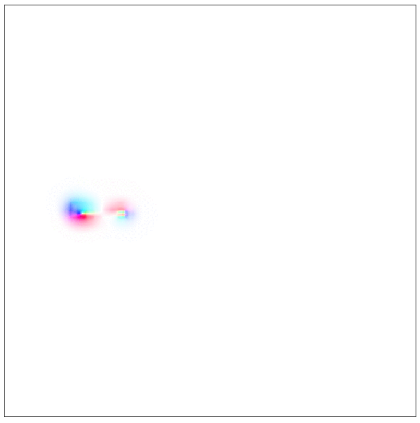

Figure 4.3 – Estimated motion field $\mathbf{w}_{\mathcal{M}_1}$.

Method $\mathcal{M}_2$ expands the observation vector with the distance map ϕ^O. The values $\phi^O(t_i)$ are displayed on Figure 4.4. This additional component ϕ^O provides information on the whole domain and the resulting estimation of motion is displayed on Figure 4.5.

These experiments highlight the fact that the variable ϕ^O is a pertinent information for the image assimilation method. However, the distance map ϕ^O includes the assumption that motion on the whole image domain corresponds to the motion of the boundary of the structures. In this experiment, as only one structure is present on the data, using ϕ^O is similar to an extrapolation of

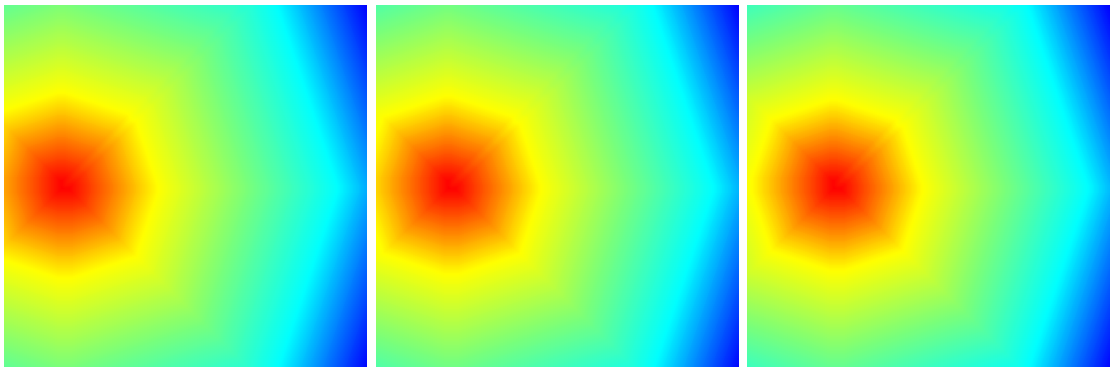

Figure 4.4 – $\phi^O(t_i)$ computed for the boundaries of Figure 4.2.

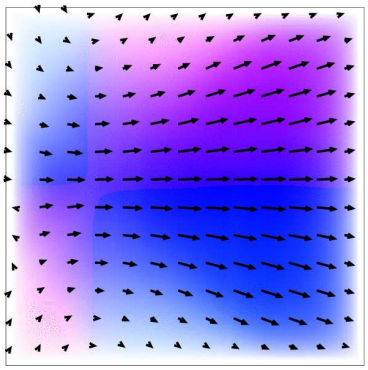

Figure 4.5 – Estimation result $\mathbf{w}_{\mathcal{M}_2}(0)$.

the motion values of the boundary to the whole domain Ω.

4.1.2 Images with texture

The model $\mathbb{M}$ of Method $\mathcal{M}_1$, composed by Equations (4.1, 4.2), is now integrated from the initial conditions displayed in Figure 4.6. The synthetic image has been created in order to display a structure, the central square, and to imitate **S**ea **S**urface **T**emperature (**SST**) images, with values between 21°C and 23°C. A sequence of five snapshots, $I^O(t_i)$ for $i = 1 \dots 5$, is extracted from the set of images resulting from the temporal integration of $\mathbb{M}$. These data are shown on Figure 4.7. Contours are first computed on the image snapshots $I^O(t_i)$ with the Deriche filter and displayed in red on the top of Figure 4.8 at date t_1, t_3 and t_5. Then, the distance maps $\phi^O(t_i)$ to these contours are calculated and displayed on the bottom of Figure 4.8 for the same dates.

First, the image components $I_{\mathcal{M}_1}(t_i)$ and $I_{\mathcal{M}_2}(t_i)$, respectively obtained by Methods $\mathcal{M}_1$ and $\mathcal{M}_2$, are compared to the image snapshots $I^O(t_i)$. They are looking almost identical and their correlation measure is over 0.99 for both Methods $\mathcal{M}_1$ and $\mathcal{M}_2$.

A similar comparison can be done with $\mathbf{X}_{\mathcal{M}_2}(t_i)$ regarding $\phi_{\mathcal{M}_2}(t_i)$. At dates t_i, the region of positive values of $\phi_{\mathcal{M}_2}$, corresponding to the inside of the tracked structure, is displayed in white and compared to the contour point, in black, computed on snapshots I^O, on Figure 4.9.

The simulation, that provides the image snapshots $I^O(t_i)$, also provides the ground truth of the motion field $\mathbf{w}_{GT}(0)$. This allows to perform statistics on the discrepancy between the estimated

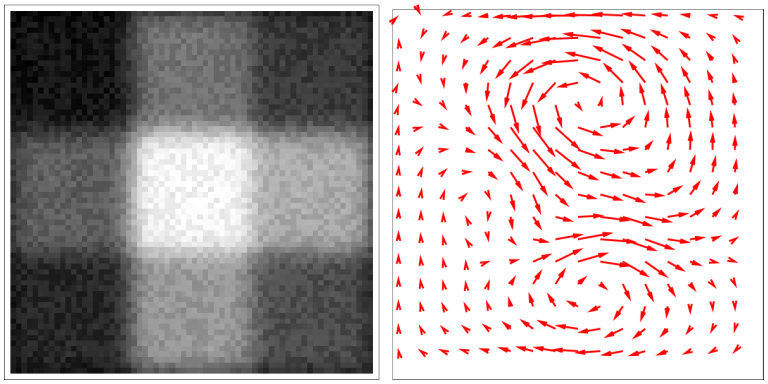

Figure 4.6 – Initial image and motion fields

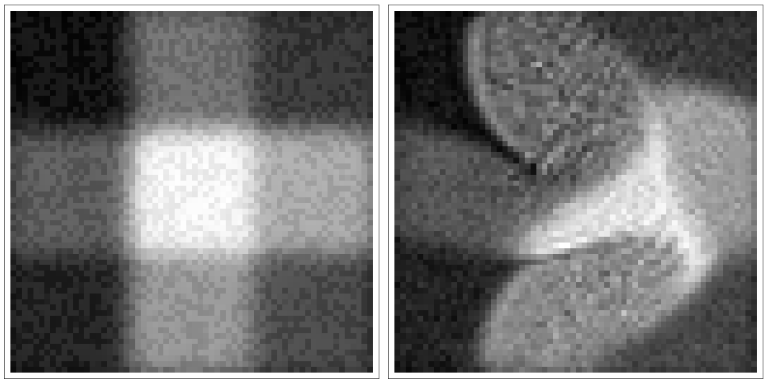

Figure 4.7 – Image snapshots at t_1 and t_5.

motion and the ground truth. Two statistical criteria are considered:

- The average of the relative norm errors, computed by:

$$\mathcal{E}_{||.||} = \frac{1}{\mathrm{N}_\Omega} \sum_{\mathbf{p}\in\Omega} \frac{abs(||\mathbf{w}_{GT}(\mathbf{p},0)|| - ||\mathbf{w}_{\mathcal{M}_i}(\mathbf{p},0)||)}{||\mathbf{w}_{GT}(\mathbf{p},0)||} \tag{4.4}$$

  N_Ω denotes the number of pixels in the discrete domain Ω.

- The average of the angular errors, computed by:

$$\mathcal{E}_\theta = \frac{1}{\mathrm{N}_\Omega} \sum_{\mathbf{p}\in\Omega} abs\left(\theta_{GT}(\mathbf{p},0) - \theta_{\mathcal{M}_i}(\mathbf{p},0)\right) \tag{4.5}$$

  where $\theta(\mathbf{p},0)$ is the angle formed by $\mathbf{w}(\mathbf{p},0)$ and a reference direction chosen as $\begin{pmatrix}1\\0\end{pmatrix}$.

The estimation obtained with Method $\mathcal{M}_1$ leads to an average of the relative norm errors, $\mathcal{E}_{||.||}$, of 11 percents while the average of the angular errors, $\mathcal{E}_\theta$, is of 1.4 degrees. For Method $\mathcal{M}_2$, the average error is around 9 percents in norm and less than 1 degree in orientation.

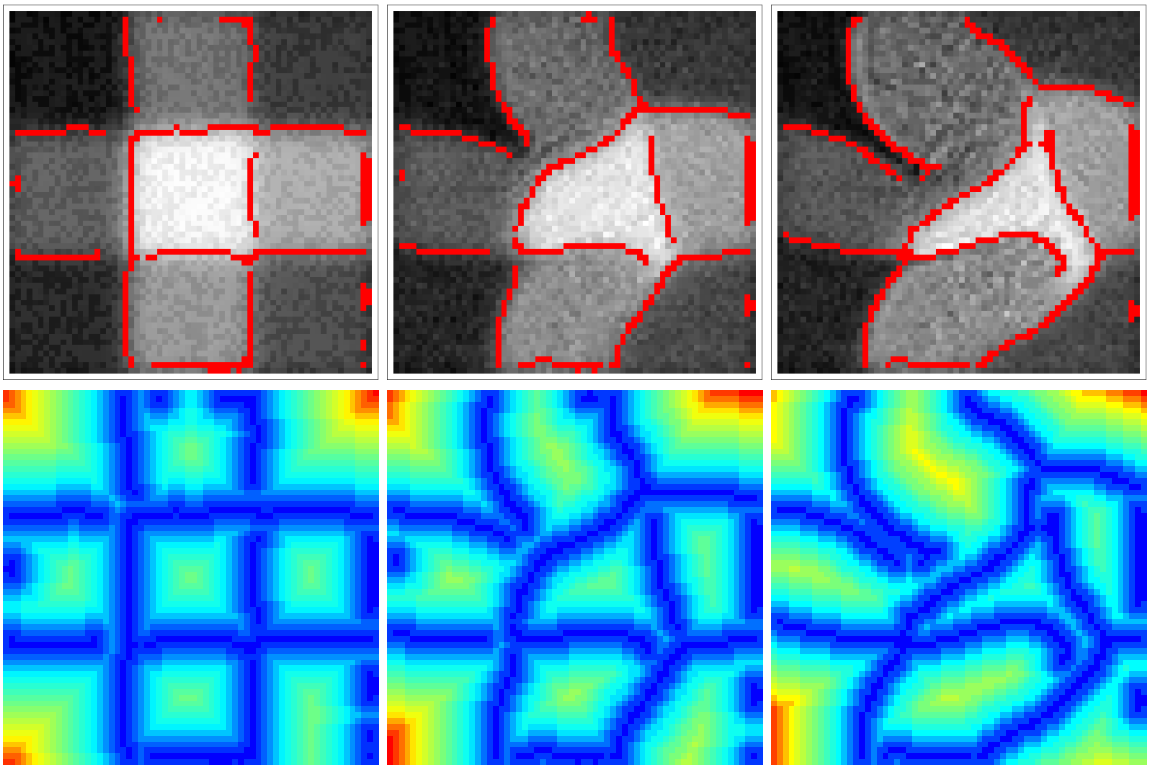

Figure 4.8 – Top line: Contour points, in red, superposed to $I^O(t_i)$ at t_1, t_3 and t_5. Bottom line: distances to these contours.

The estimated motion fields, $\mathbf{w}_{\mathcal{M}_1}(0)$ and $\mathbf{w}_{\mathcal{M}_2}(0)$, and the ground truth $\mathbf{w}_{GT}(0)$ are displayed on Figure 4.10 with the colored motion representation. It can be seen that, even if the statistics show only little improvement in the estimation obtained by $\mathcal{M}_2$ compared to $\mathcal{M}_1$, the estimation $\mathbf{w}_{\mathcal{M}_2}(0)$ better recovers the structures of the motion ground truth $\mathbf{w}_{GT}(0)$.

4.1.3 Importance of the reinitialization module

This subsection further investigates the estimation obtained with Method $\mathcal{M}_2$. It analyzes the use of the structures representation ϕ. The aim is to demonstrate the necessity of including the reinitialization module in the Image Model $\mathbb{M}$ of Method $\mathcal{M}_2$. For that purpose, we compare the results of the estimations obtained with and without this reinitialization module. First, the model $\mathbb{M}$ of Method $\mathcal{M}_1$, composed by Equations (4.1, 4.2), is integrated with the initial conditions displayed on Figure 4.11. The initial image has been acquired in the infrared domain by the satellite Meteosat Second Generation. The idea is to use a real satellite image with a synthetic motion field as initial conditions, in order to generate a sequence of snapshots, which are similar to real data while keeping the advantage of having the ground truth on motion. A sequence of eighteen snapshots, $I^O(t_i)$ for $i = 1$ to 18, is extracted from the image data resulting from the integration of $\mathbb{M}$. Then contours $C^O(t_i)$ are computed on images $I^O(t_i)$ and the distance maps $\phi^O(t_i)$ are derived.

Method $\mathcal{M}_2$ is then applied, resulting in a motion estimate $\mathbf{w}_{\mathcal{M}_2}(0)$. Statistics are computed on the discrepancy between the motion field $\mathbf{w}_{\mathcal{M}_2}(0)$ and the ground truth $\mathbf{w}_{GT}(0)$. They are given

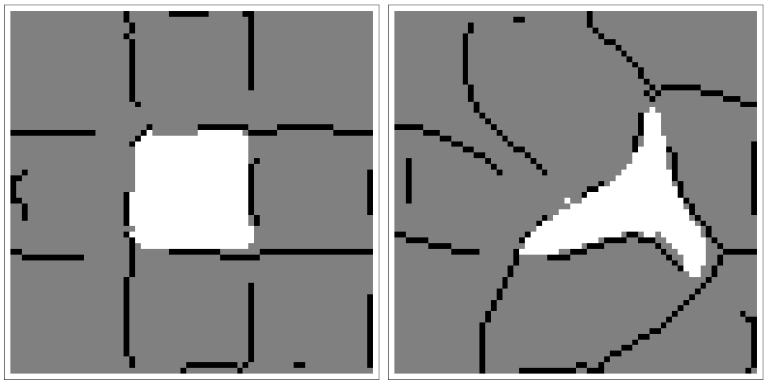

Figure 4.9 – Comparison of $\phi_{\mathcal{M}_2}$ (positive values in white) and contour points (in black). Left: t_1, Right: t_5.

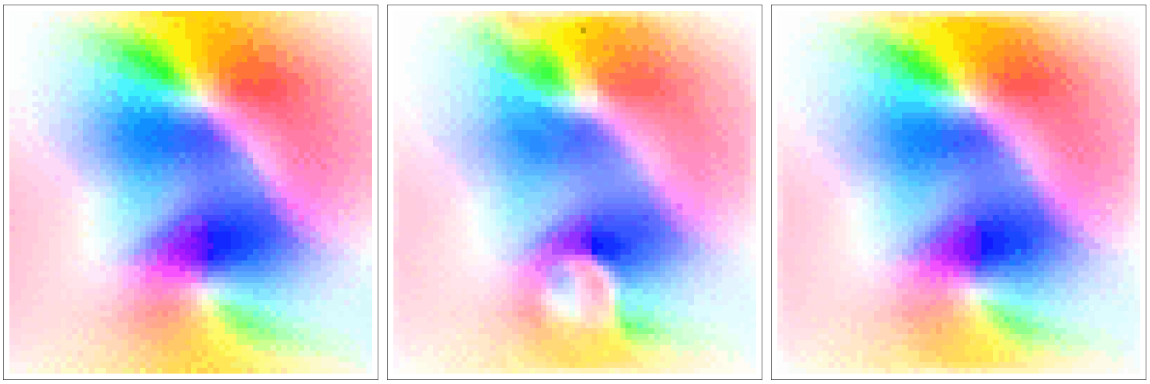

Figure 4.10 – From left to right: Ground truth, result of $\mathcal{M}_1$ and result of $\mathcal{M}_2$. At initial date.

in the first column of Table 4.1. The second column provides statistics obtained if the motion is

	with Reinit	without Reinit
Relative norm (in %):	12	21
Angular (in degrees)	5.5	8.5

Table 4.1 – Statistics on the errors between Method $\mathcal{M}_2$ ($\mathbf{w}_{\mathcal{M}_2}$) and ground truth ($\mathbf{w}_{GT}$).

estimated by Method $\mathcal{M}_2$, but without including the reinitialization module. It can be seen that errors are clearly reduced thanks to the reinitialization module. If the reinitialization module is not used in the Image Model $\mathbb{M}$ of $\mathcal{M}_2$, the advected function ϕ is no more a distance map, as discussed in Subsection 2.3.1 of Chapter 2. Therefore, the comparison between ϕ_i and ϕ_i^O makes less sense. It leads, in the best case, to no improvement compared to the estimation $\mathbf{X}_{\mathcal{M}_1}$ obtained by Method $\mathcal{M}_1$ and, in the worst case, it will deteriorate the estimation $\mathbf{X}_{\mathcal{M}_1}$.

4.1.4 Comparison with the state-of-the-art

The experimental setting of Subsection 4.1.3 is applied again for comparing our methods with those of the state-of-the-art. The real satellite acquisition of MSG2, displaying a convective cell, is

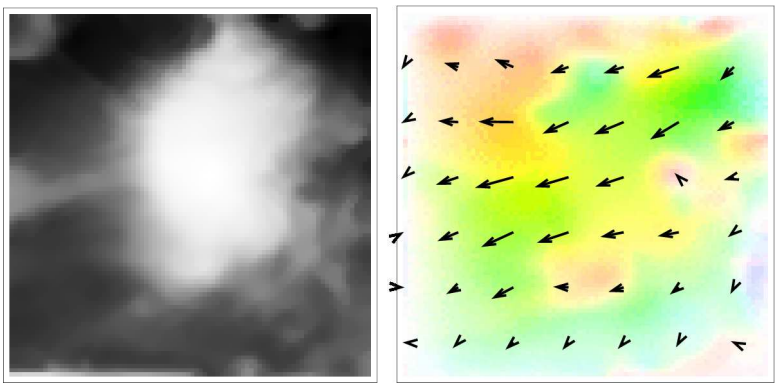

Figure 4.11 – Left: initial image. Right: initial motion field.

used with the synthetic motion field, displayed on Figure 4.11 in order to generate the sequence of snapshots.

Before comparing our estimations with the state-of-the-art, let first compare the results obtained by Methods $\mathcal{M}_1$ and $\mathcal{M}_2$. The motion results are displayed on Figure 4.12: on the left the ground truth, in the middle the result obtained by $\mathcal{M}_1$ and on the right the one obtained by $\mathcal{M}_2$. The

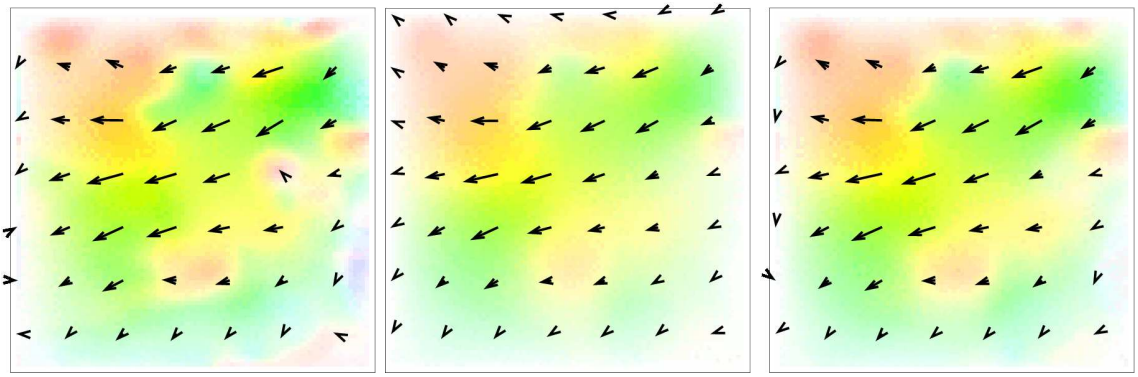

Figure 4.12 – From left to right: $\mathbf{w}_{GT}(0)$, $\mathbf{w}_{\mathcal{M}_1}(0)$ and $\mathbf{w}_{\mathcal{M}_2}(0)$.

estimation $\mathbf{w}_{\mathcal{M}_1}(0)$ obtained by Method $\mathcal{M}_1$, compared with $\mathbf{w}_{GT}(0)$, leads to a relative norm error of 25 percents (Equation (4.4)) and an average angular error of 9.5 degrees (Equation (4.5)). Method $\mathcal{M}_2$ reduces these errors, respectively, to 5.5 degrees and 12 percents.

Method $\mathcal{M}_2$, which provides the best result, thanks to the structures representation ϕ, is then compared with the optical flow method from Sun et al. [11], denoted $\mathcal{M}_{OF}$. This method has been chosen as it is one of the method leading to the lowest error statistics on the Middlebury benchmark [13].

Let remark that a direct comparison of these two approaches is not possible, as Method $\mathcal{M}_2$ provides a motion field at each time step of the studied temporal interval, while the optical flow method provides a unique motion field in-between two consecutive image snapshots. Quantification of results is then achieved by comparing the trajectories of characteristic features, which are transported by these two motion fields. For illustration purposes, in this document, we chose 8 points on the boundary of the structure and compute their trajectories. Figure 4.13 displays the position

of these 8 points at the final date. On the image, the red points are used to show the ground truth positions on the last image, the blue points are used for the positions obtained by Method $\mathcal{M}_2$ and green color displays the positions resulting from Method $\mathcal{M}_{OF}$. Ellipses are visualized according to the color of the best estimation result (blue or green), gray being used when the estimations have a similar quality.

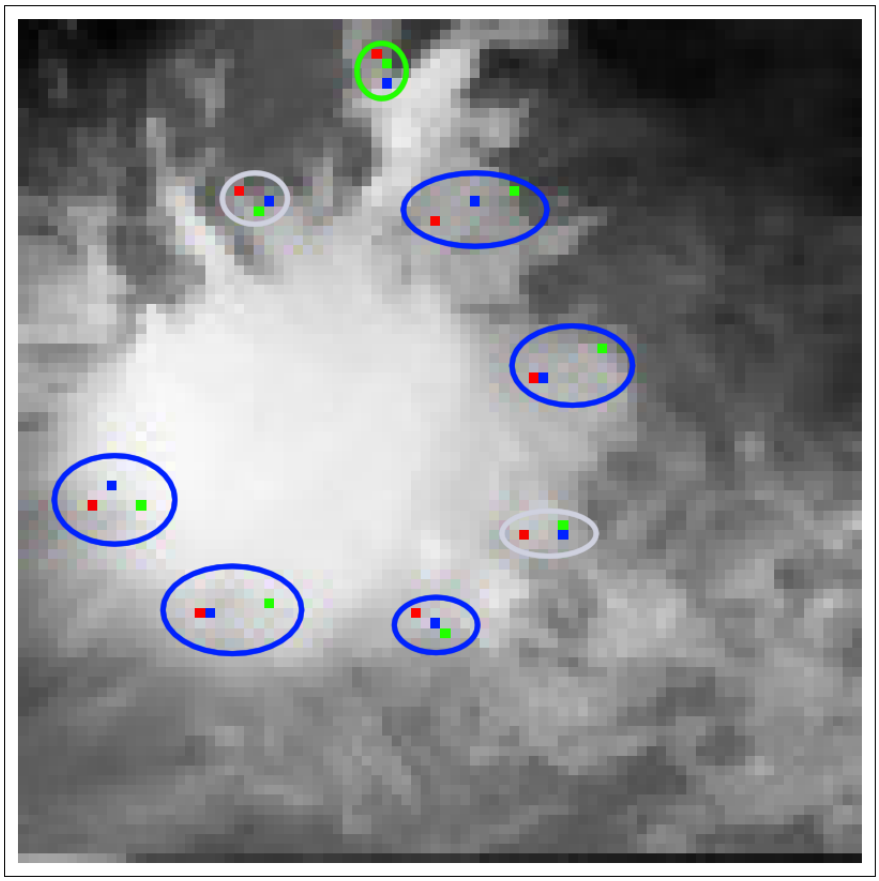

Figure 4.13 – Comparison of the ground truth (*in red)* and the estimation results obtained by an optical flow algorithm $\mathcal{M}_{OF}$ (*in green)* and our estimation method $\mathcal{M}_2$ (*in blue)*.

The mean of the distances between the ground truth positions and the estimated ones, computed on more than 200 characteristic features, is around 2.2 pixels with our approach, whereas it is above 4.6 pixels for the optical flow method $\mathcal{M}_{OF}$.

Having assessed, through twin experiments, the ability of our image assimilation methods to correctly estimate motion, it is now possible to apply them on real acquisitions.

4.2 Experiments on real images

4.2.1 Satellite images from clouds

The assimilation Methods $\mathcal{M}_1$ and $\mathcal{M}_2$ are now applied on a Meteosat Second Generation sequence of 18 acquisitions, from which four images are displayed on Figure 4.14. These images are acquired

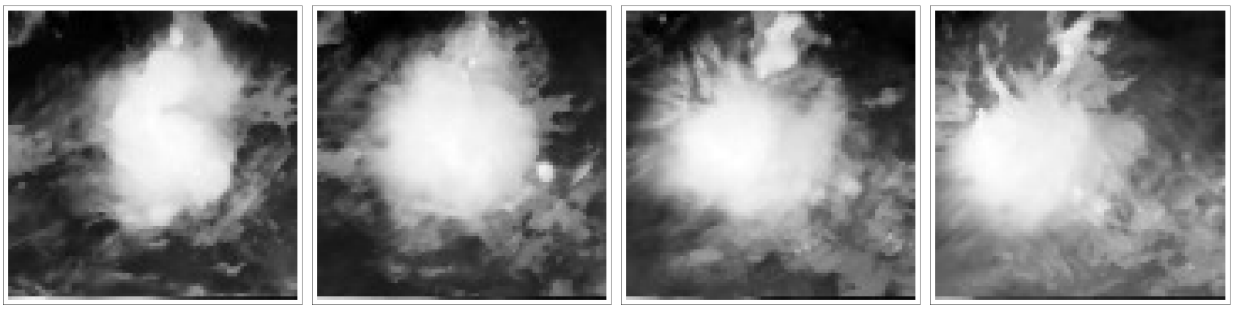

Figure 4.14 – From left to right: images of a tropical cloud in the infrared domain.

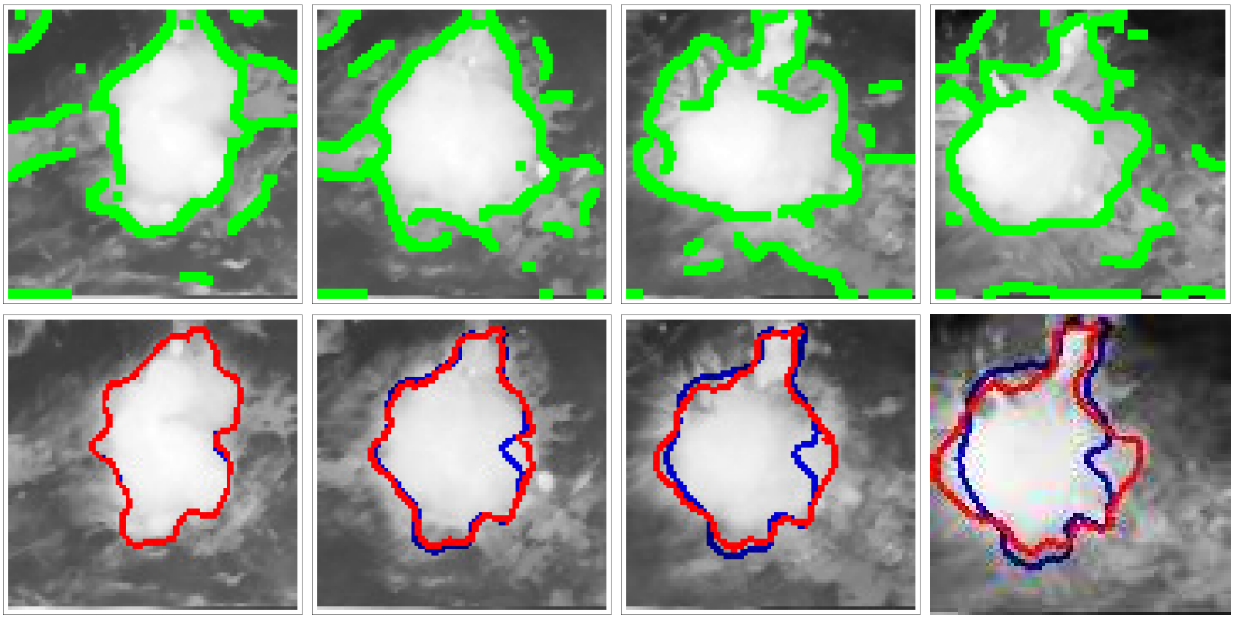

Figure 4.15 – Top: Contours on image acquisitions. Bottom: red is the result obtained by $\mathcal{M}_2$ with distance map ϕ, blue is obtained by $\mathcal{M}_1$ without ϕ.

every 15 minutes in the infrared domain with a 5 kilometer resolution.

The first line of Figure 4.15 displays the contour points, used to compute the distance maps $\phi^O(t_i)$, involved in the observation vectors. On the second line, the blue curve corresponds to the advection of the cloud boundary by the motion field estimated with Method $\mathcal{M}_1$, without ϕ in $\mathbf{X}$. The red curve is the distance map ϕ estimated by $\mathcal{M}_2$. The image assimilation is computed on the whole image domain Ω, but Figure 4.15 only focuses on the cloud structure. As it can be seen, including constraints on ϕ allows to improve the accuracy of tracking the structure. It shows that the motion field estimated by Method $\mathcal{M}_2$ correctly assesses the displacement of the cloud along the whole sequence.

4.2.2 Ground radar data

Chapter 5 focuses on the nowcasting of flash floods thanks to the short term forecast of precipitation quantities. The proposed operational system relies on two components: an estimation phase, using

the assimilation Method $\mathcal{M}_1$ or $\mathcal{M}_2$, and a forecast phase. For illustrating the results of image assimilation, some results of the estimation phase are given in the current section, while Chapter 5 will focus on the description of the operational system and on the analysis of the results obtained by the forecast phase.

Figure 4.16 displays three consecutive acquisitions obtained by a ground radar. These data were acquired over Clermont-Ferrand every three minutes on April 24^{th} of 2014 from 5:33 PM to 5:39 PM. For a detailed description of these acquisitions, the interested Reader should refer to Chapter 5. These three acquisitions display a rain cell entering the radar acquisition domain from the upper

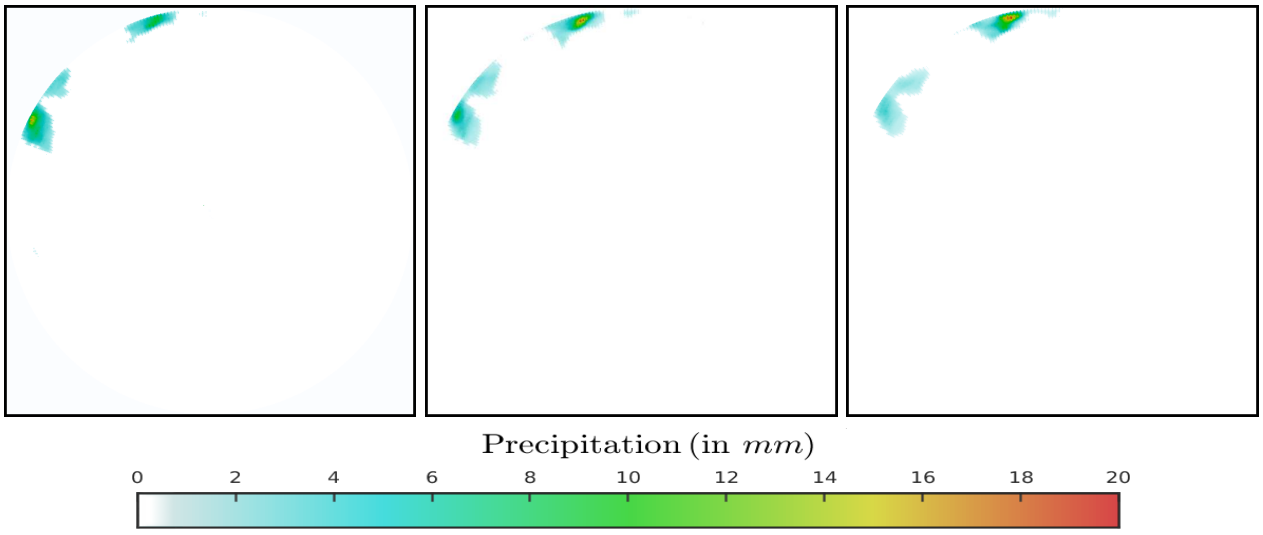

Figure 4.16 – Observations $\mathbf{Y}_1$,$\mathbf{Y}_2$ and $\mathbf{Y}_3$.

left corner. At 5:33 PM, the precipitation cells are progressively arriving over Clermont-Ferrand and will lead to a storm with precipitation rates up to 100 mm/h.

We apply both Method $\mathcal{M}_1$ and Method $\mathcal{M}_2$ on these radar acquisitions. The observation vector is composed, for Method $\mathcal{M}_1$, of the acquisition $I^O(t_i)$. The corresponding observation error covariance matrix R_I is chosen diagonal. It describes the acquisition noise as corresponding to ten percent of the measured values. When estimating motion with Method $\mathcal{M}_2$, the structures representation $\phi^O(t_i)$, obtained by applying a threshold process to the acquisitions, is added to the observation vectors. This distance map models the regions of high precipitation rates. The observation covariance matrix associated to the distance map, R_ϕ, is also chosen diagonal. The observation covariance matrix R_I associated to the image component and used by Method $\mathcal{M}_2$, is the same as the one used for Method $\mathcal{M}_1$.

For each estimation method, the background includes a null motion field $\mathbf{w}(0)$ and the first image acquisition $I^O(0)$. The methods do not constrain the initial value $\mathbf{w}(0)$. The background error covariance matrix B only includes the covariance component B_I for Method $\mathcal{M}_1$. An additional component B_ϕ is added for Method $\mathcal{M}_2$. As it has been done with R_I, the background covariance matrix B_I is taken diagonal with values corresponding to the acquisition noise. When using Method $\mathcal{M}_2$, the background on $\phi(0)$ is initialized by $\phi^O(0)$ and the background covariance matrix B_ϕ gets the same values than R_ϕ.

This experimental setting leads to the estimated motion fields displayed on Figure 4.17. The

motion fields are represented with both arrows and colors. It can be seen, on this figure that the

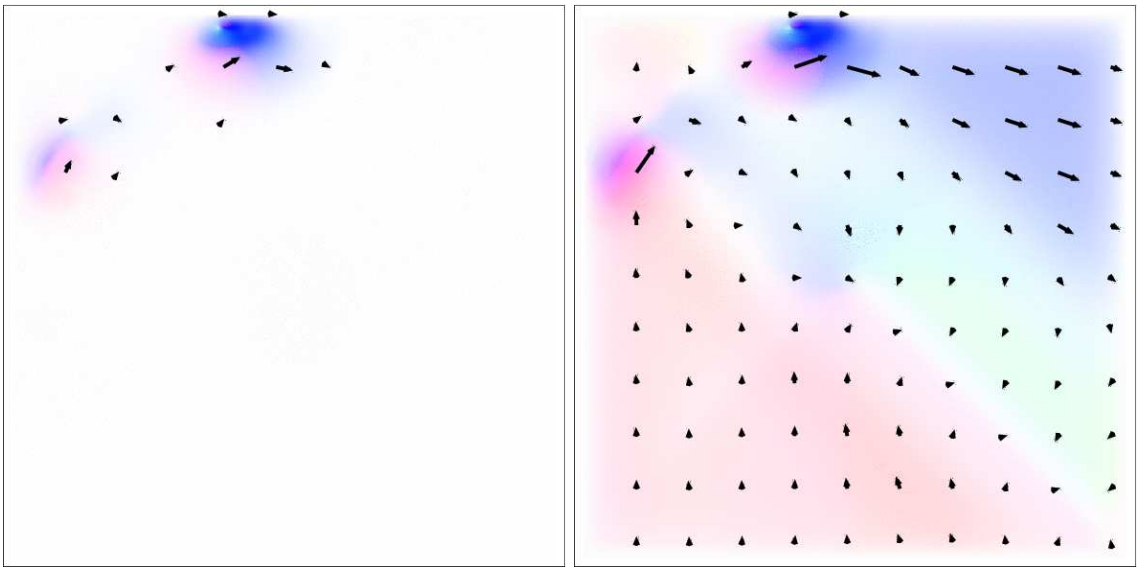

Figure 4.17 – Estimation result $\mathbf{w}_{\mathcal{M}_1}$ on the left and $\mathbf{w}_{\mathcal{M}_2}$ on the right.

image assimilation with Method $\mathcal{M}_1$ estimates motion only on regions for which the radar measures are non null. As stated before, if the image data do not display any displacement, no motion can be estimated in this case. However, Method $\mathcal{M}_2$ is using the function ϕ representing, at each pixel $\mathbf{p}$, the distance to the closest point on the boundary of the high precipitation cells. This distance map ϕ adds information on the whole domain Ω. This allows Method $\mathcal{M}_2$ to get an estimation result at each pixel $\mathbf{p}$. The regularization terms also help to enlarge the region where motion has a non null value, as they impose smoothness of the estimation and therefore prevent high values of the spatial gradient of the estimated motion. However, these regularization terms also tend to diminish the norm of the estimation. Therefore, a compromise has to be found and a learning phase is required in order to obtain an optimal parametrization of the regularization terms for the nowcasting system.

The operational system relies on a sliding window implementation of the estimation/forecast phases. This approach, well known and widely used in the image processing and data assimilation communities, is described in details in Subsection 5.2.2 of Chapter 5. In order to illustrate the results obtained with an on-line setting, another estimation window is displayed, with data acquired every five minutes on August 8^{th} of 2014 from 4:00 PM to 4:10 PM. These radar images are shown on Figure 4.18. The only modifications for this experiment, compared to the first presented one, concerns the motion background, which is equal to the estimation on the previous window, as explained in Subsection 5.2.2 of Chapter 5. The motion result $\mathbf{w}_{\mathcal{M}_2}$ is displayed on Figure 4.19. It can be seen on this figure that motion $\mathbf{w}_{\mathcal{M}_2}$ is estimated even in regions of null precipitation rates on the radar data. This is particularly visible on the left part of the domain. This property is resulting from the sliding window procedure and the utilization of the distance map ϕ. The use of the last motion result as background for the next window also adds a temporal coherence to the estimation.

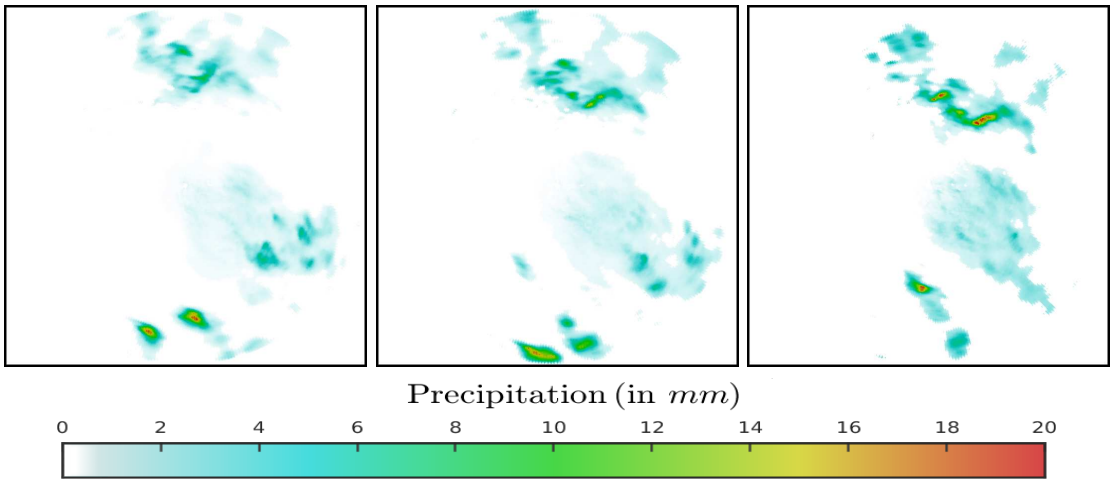

Figure 4.18 – Observations $\mathbf{Y}_{40}$,$\mathbf{Y}_{41}$ and $\mathbf{Y}_{42}$.

4.3 Conclusion

An objective validation of the estimated motion fields is nearly impossible if using real data in operational conditions. Tests have first been done with twin experiments and have shown that the motion estimation Methods $\mathcal{M}_1$ and $\mathcal{M}_2$ are able to recover the motion field underlain in the evolution of an image sequence. Some cases are more favorable to Method $\mathcal{M}_2$, for instance in the total absence of texture on the images used for the observations. However, in more realistic tests, both estimation methods behave properly.

In real applications, without any ground truth, the quality of the estimation can hardly be evaluated. Comparison of the image function I of the state vector and the acquisition I^O can give an intuition of the quality of the estimation. A comparison of the contour points, computed on the acquisition I^O, and of the structure boundary, that is advected by the estimated motion field, has been given to illustrate some differences between Methods $\mathcal{M}_1$ and $\mathcal{M}_2$.

Regarding the experiments for which the image acquisitions contain poor information about motion, i.e. no texture or large regions without moving structures, Method $\mathcal{M}_2$ outperforms the estimation abilities of Method $\mathcal{M}_1$. In cases for which the information about motion is dense, both methods perform similarly. The choice has therefore been made to consider Method $\mathcal{M}_2$ as estimation method for the nowcasting operational system presented in Chapter 5. Nevertheless, a few more comparisons between Methods $\mathcal{M}_1$ and $\mathcal{M}_2$, regarding their associated forecast results, will be given in the next chapter.

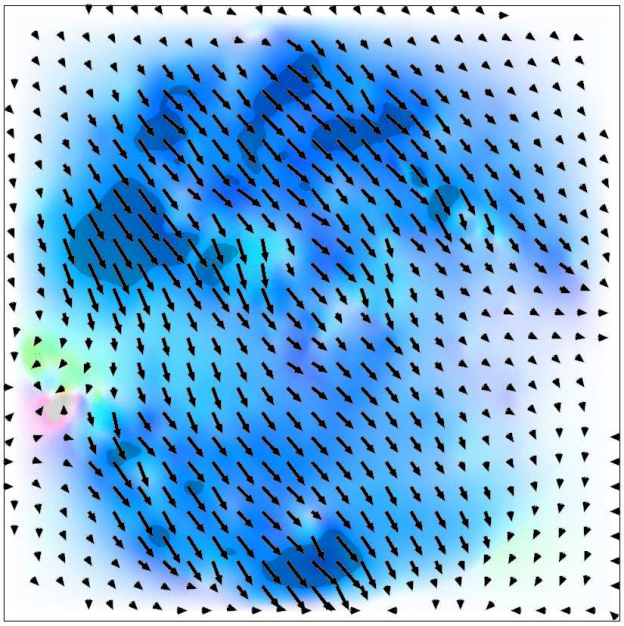

Figure 4.19 – Motion result for the 40^{th} temporal window.

Chapter 5

Assimilation of radar reflectivity for nowcasting of precipitations

5.1 Foreseen operational use

The aim of this chapter is to describe the design of an operational system dedicated to the short term forecast of precipitations. Providing accurate estimation of rain is, for instance, crucial to prevent flash flood consequences and ensure the security of populations, for airport management and for agricultural monitoring. Location of regions with high precipitation rates and their associated quantities have to be accurate in order to be used by decision-making procedures, for avoiding danger and unnecessary costs. The application of preventive measures requiring a fixed and predetermined duration, the nowcasting system must take this delay into account and provide forecast enough time in advance.

These preliminary remarks lead to two major constraints while designing the system. First, forecasts have to be updated as soon as new acquisitions are available in order to detect the formation of new clouds and the appearance of new rainy events. This constraint is mandatory if the forecasting system is only based on image acquisitions, as modeling the cloud formation would additionally require the use of physical laws. Second, the temporal horizon of accurate forecasts must be high enough for being able to apply appropriate preventive measures.

Regardless of the mathematical method that is applied for forecasting precipitations, one has to choose the acquisitions to be used as input to the system. Several acquisition systems are at hand, either satellite radars, as the GPM Core Observatory[1] launched in 2014, or ground based networks, for instance the french global radar network Aramis [51] or local radars. The differences between these systems are mainly the temporal frequency and the spatial resolution of the acquisitions. The Nasa satellite GPM has a resolution of five kilometers for an acquisition width up to 885 kilometers. However, the sensor sends its data once per orbit, which means every ninety minutes. Consequently, these acquisitions can not be used for applications like the precipitations nowcasting up to one hour. The french global network Aramis is composed of 24 radars generating a mosaic of the reflectivity measures over France. It has an acquisition rate of 5 minutes. Such a short temporal period renders the nowcasting possible, but the Aramis sensors have a resolution of 1.5 kilometers, which, for a precise forecasting of precipitations over a city, is considered as a low resolution. The ground-based radar images of the rain reflectivity used in this chapter are provided by the company

[1] http://www.nasa.gov/mission_pages/GPM/spacecraft/index.html

Weather Measures[2], located in Clermont-Ferrand. The radars have an acquisition rate of less than 5 minutes and a resolution of 150 meters. The acquisitions used for this application are extensively described in Section 5.2.

The discussion is illustrated with the city of Clermont-Ferrand, France, displayed on a map of France on the left of Figure 5.1. The right image of Figure 5.1 shows a zoom in of the satellite view of Clermont-Ferrand, with the flooding risk areas colored in blue.

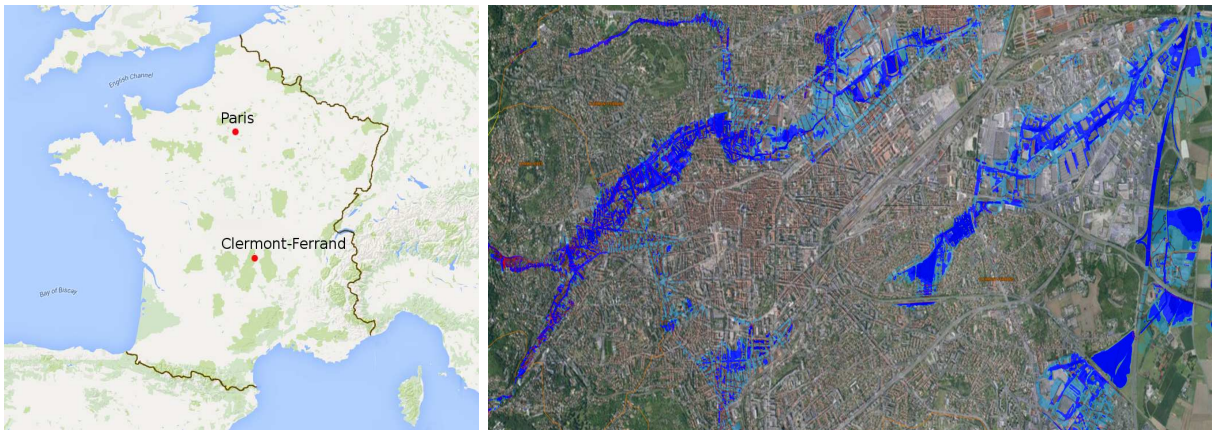

Figure 5.1 – Clermont-Ferrand displayed on a map of France (*on the left)* and the flooding risk areas in Clermont-Ferrand (*in blue on the right image, credit to the "Ministère du développement durable".)*

The usual temporal horizon of the operational forecasts, which are required in this context, ranges from thirty minutes to two hours. The nowcasting system to be designed is composed of two major components:

- an estimation component that is based on the acquisitions and estimates the state of the atmosphere;
- a forecast phase that uses the previous estimation in order to forecast the future state of the atmosphere.

As the nowcasting system concerns the short-term horizon, the forecast must be available a few minutes after an acquisition and at least before the next acquisition. The forecast must also be updated with the same time period than the radar sensor. Numerical Weather Prediction (NWP) methods are based on physically accurate mathematical models of the atmosphere. However, they require a high computational time (usually in the order of hours) for the estimation phase and therefore the acquisitions, on which the forecast relies, have been acquired long ago. This delay explains why NWP methods are usually unable to forecast with accuracy the localized events, which appear promptly.

An alternative that is proposed in the literature concerns the image-based systems. The estimation component relies on image processing methods that compute motion from an image sequence. The duration required by this estimation component is much shorter than the one of NWP methods. For a survey of image-based methods, the interested Readers may consider the papers of Mecklenburg et al. [52] and Ebert et al. [53]. The methods presented in these references detect the clouds on

[2] `http://www.weather-measures.fr/`

the image domain and, then, by processing consecutive images, determine their displacements from one image to the next one. These displacement fields are further used for forecasting the future position of the clouds and the rain quantities. For instance, the nowcasting system called CALAMAR, whose methods are proposed in the thesis of Neumann [54], works as follow. First, a segmentation of the regions with high precipitation rates is applied on two consecutive acquisitions. In a second step, the system labels each region and computes its center and a surrounding polygon. In a third step, a process matches the regions of the first image with their corresponding region in the second image. Each region from the first acquisition being linked with one on the second acquisition, a motion value and a growth rate are then estimated. Last, these motion and growth rate values are used for generating the forecast. The nowcasting platform TITAN, described by Dixon and Wiener in [55], relies on a similar procedure. As these systems, CALAMAR and TITAN, depend on image data, the quality of their results is highly dependent on the quality of the acquisitions. If images are corrupted by noise, the methods will fail to accurately predict the displacements and consequently will fail to nowcast the rain quantities.

In order to be less impacted by the image quality, we decided to design a new estimation component, based on an image assimilation approach with an image model $\mathbb{M}$. As it will be seen, the model $\mathbb{M}$ describing the temporal evolution and heuristics on the clouds displacement, is able to minimize the impact of the acquisition noise on the estimation result. As it is used by the estimation component, the Image Model is denoted $\mathbb{M}_e$ in the remaining of the chapter. To estimate the motion field that optimally fits to the acquisitions under the constraint of the evolution laws of $\mathbb{M}_e$, the image assimilation method is based on the 4D-Var technique. The estimation method and its major components were fully described in Chapter 1 and Chapter 2. Both approaches, with and without structure components, respectively named Methods $\mathcal{M}_2$ and $\mathcal{M}_1$, are applied to the rain nowcasting issue in the remaining of the chapter.

Having described the estimation component, one has to design the forecast component. It relies on a forecast model $\mathbb{M}_f$ that can be identical, or not, to the estimation model $\mathbb{M}_e$. The choice of $\mathbb{M}_f$ is dependent on the temporal horizons that are foreseen in the operational context. Under the collaboration that is conducted with the company Weather Measures, the estimation is obtained from acquisitions acquired during a temporal interval of 10 minutes, while the forecast horizon ranges from 30 minutes to 2 hours. In that context, the heuristics used to define the forecast model $\mathbb{M}_f$ are different from the one applied to define the estimation model $\mathbb{M}_e$. The forecast component is fully described in Section 5.3.

Validation of the methods and results is crucial for an operational use of the system. Different validation indexes are proposed and described in Section 5.4. Having more than three hundred acquisitions taken over five days during important rain events, validation is applied on a large number of forecast results. However, the validation process compares the forecasts to the future acquisitions, which are considered as the truth. Some issues related to the acquisitions made by ground-based radar, which are described in Section 5.2, can considerably affect the outcomes of the validation tools. Some perspectives will be given in order to solve these effects.

This chapter is divided in three sections. Section 5.2 describes the whole operational context, from the acquisition to an optimal use of the estimation component in a so-called sliding window framework. Section 5.3 focuses on the description of the forecast model $\mathbb{M}_f$ used to produce a rain quantity estimation at short time horizon. Last, Section 5.4 displays the results obtained when applying the whole system on five different rainy events of summer 2014 and gives statistics about the performances.

5.2 Operational context

This section aims to describe the whole operational context in which the nowcasting system should be incorporated. First, an extensive description of the acquisitions is given in Subsection 5.2.1. This subsection describes how information on the precipitation rate is measured by the radar and some problems inherent to these acquisitions. Then, Subsection 5.2.2 recalls some notations used in the chapter and discusses the framework of sliding windows. Using this framework for image assimilation allows the temporal coherence of the resulting estimation.

5.2.1 Acquisitions

The acquisitions used for this study have been provided by the company Weather Measures. They are obtained with the X band radar WR10X[3], which is displayed on Figure 5.2. The radar sensor

Figure 5.2 – The WR10X radar used for the nowcasting application.

measures the reflectivity, which corresponds to the fraction of incident electromagnetic energy that is reflected at interfaces. This section will briefly discuss some problems inherent to the functioning of the radar and describes some possible solutions for processing such data. The actual pre-processing used in our operational system is then described.

The electromagnetic waves emitted by the radar are reflected by the rain. The ratio of reflectivity corresponds to the measures given on the acquisitions. However, the reflection also occurs on the earth surface, vegetation, buildings or mountains. This causes artificial echoes on the image data. The phenomenon is called ground clutter, it is particularly visible in the absence of precipitation, and wrongly interpreted as a high precipitation rate. As, the ground clutter is mostly stationary, the processing of temporal sequences leads to a null motion estimation at such places. This effect may be suppressed by a pre-processing of the acquired measures. The idea is to measure the response from the ground on a sunny day and subtract this response on each new acquisition in order to obtain the reflectivity values that are only due to the rain drops.

The radar emits electromagnetic waves and, while going through the interface of different media, a part of them is reflected. This reflected part of the electromagnetic waves is measured by the radar and gives information on the precipitations. However, during the propagation, the waves, by

[3] `http://www.eldesradar.com/WR10X.htm`

traveling trough different media, are attenuated. Thus, the corresponding reflectivity values and the associated precipitation rates are underestimated.

A method that allows decreasing the impact of the attenuation is proposed in the paper of Sempere et al. [56]. Their solution consists in analyzing the ground clutter linked to mountains, or any other objects in the radar acquisition range, in order to correct the acquired values.

Attenuation impacts every acquisition, by nature, and any medium crossed by the electromagnetic waves provokes it. The first one is the dome over the radar, called the radome, which can be seen on the left image of Figure 5.2. The effect of the radome, when it is dry, is well known but the corresponding attenuation during a hail over the radar is not. In such situations, the radome is wet and the electromagnetic waves fail to propagate outside a small perimeter around the radar. This problem is illustrated on Figure 5.3 where images are displayed on a Cartesian grid. The first two images display a large region of high precipitations, which is progressively arriving over the radar. Once being over this radar, on the third image, the surface of the rainy region becomes smaller. It expands again after moving away, from the fourth image. It seems logical to consider that the

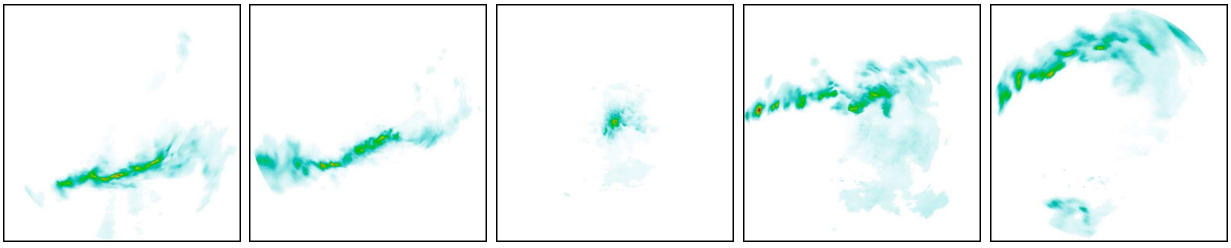

Figure 5.3 – Sequence in which the attenuation due to the rain over the radome is particularly visible.

front kept its width all over the sequence and that the radar underestimated it on the third image due to the water on the radome.

A solution for evaluating attenuation is to use a network of radars. The attenuation factor of each individual radar R would be estimated from the measures of the others. This attenuation coefficient will be obtained by looking for a temporal incoherence of the various available measures of reflectivity. In other words, the system will look for drastic changes in the values of the measured reflectivity. If the drastic change is measured by several radars, the measures are considered correct. However, if this change occurs only for a single radar, it is most likely that the cause of the change is due to attenuation. The final reflectivity value would then be obtained from a correction of the measure, accordingly to an estimation of the attenuation coefficient.

As said before, the radar sensor measures the fraction of incident electromagnetic energy that is reflected at the interface of the rain drops. In order to obtain measures that are proportional to the diameter and distribution of the rain drops, the scanning waves, sent by the radar, must follow a Rayleigh scattering law. The reflectivity measures are then further analyzed to determine the precipitation rate, using the so-called Z-R equation:

$$Z = a \times R^b \tag{5.1}$$

where the reflectivity Z (expressed in dBZ) is transformed into a precipitation rate R (in $mm \cdot h^{-1}$). This transformation is usually done with the coefficients given by Marshall and Palmer in [57] that best represent precipitations at our latitude. In that case, $a = 200$ and $b = 1.6$:

$$Z = 200 \times R^{1.6} \tag{5.2}$$

The constants a and b, used in the Z-R relation of Equation (5.1), are in fact dependent of the precipitation type and in particular if it rains or if it snows. These constants a and b are also dependent of the precipitation rate. They are dependent from whether the cells are growing or shrinking, and so on. The study from Van Baelen et al., see [58], gives a more precise description on the subject. For each case, a corresponding couple value (a, b) is given. However, the implementation of such an adaptive Z-R equation would drastically increase the computational time of the pre-processing of the radar acquisitions. Moreover, additional information on the actual state of the atmosphere would be needed for determining which parameters values should be used in Equation (5.1) at each location of the domain. This process is not affordable in the studied operational context.

Consequently, we make the choice to use the constants determined by Marshall-Palmer and described by Equation (5.2). These constants are statistically the best one for describing the rain at the latitude of the test site (Clermont-Ferrand). Figure 5.4 displays, on the left an example of reflectivity image and on the right the corresponding precipitation rate image, obtained by the Marshall-Palmer law. The images are shown according to the polar coordinates; the line index corresponds to the angle of acquisition and the column index to the distance to the radar. This polar system is the one used by the acquisition.

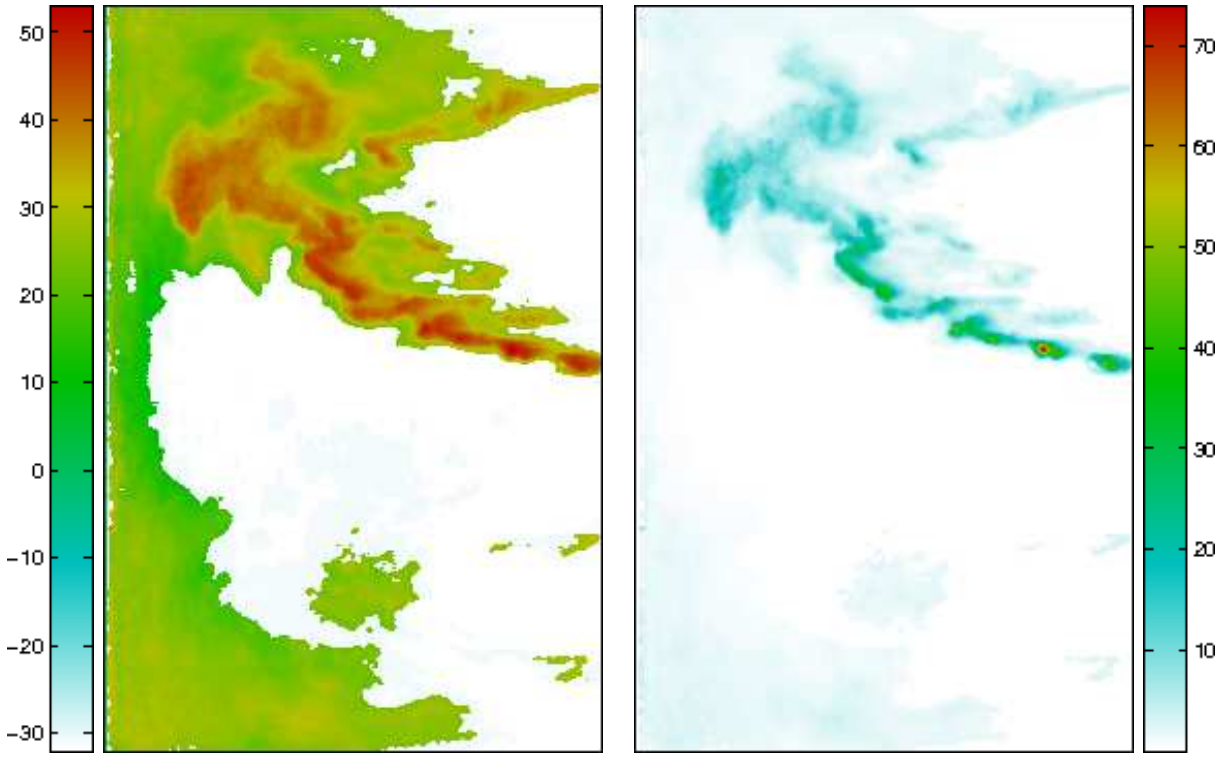

Figure 5.4 – A measured reflectivity image (*on the left)* and its corresponding precipitation rate (*on the right).*

Having given some general information on radar acquisitions, we will now describe the pre-processing that is applied in our system. The acquisitions used for the nowcasting are obtained every five minutes on some events and every three minutes on others. On the horizontal plane, data are acquired along rays: the angular resolution is one degree and each ray is composed of

240 points, with a resolution of 150 meters, starting from the radar location. As explained, the acquired values are first transformed into precipitation rates with the Mashall-Palmer relation of Equation (5.2). Then, in order to process the precipitation rates on a Cartesian grid, the polar acquisitions are projected on a 721×721 pixels grid, with a 100 meter resolution.

Figure 5.5 illustrates the pre-processing. The image on the left corresponds to the polar acquisition of the reflectivity values transformed into precipitation rates. From the polar precipitation rates is obtained the Cartesian image, which is the second image of the figure. As visible, the result of this transformation is sparse. Moreover, the farthest to the radar position the sparser is the data availability. This sparsity issue is solved as a missing data problem. A diffusion process is applied on the Cartesian grid for filling in the missing values: starting from the pixels where precipitation rates are available, these values are progressively diffused to the rest of the domain using the well known temperature diffusion equation. The image on the right of Figure 5.5 corresponds to the precipitation rate after diffusion to the grid points where acquisitions were missing.

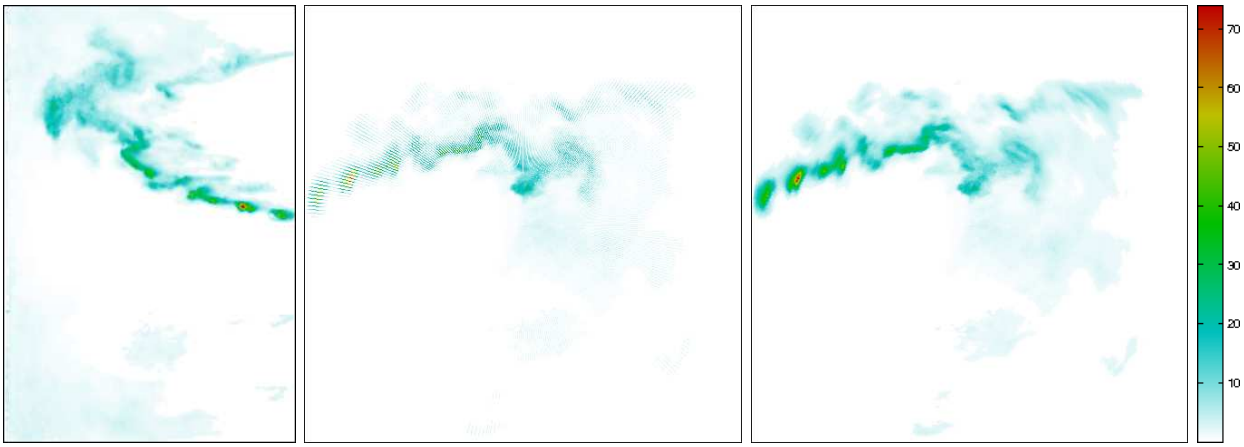

Figure 5.5 – Polar precipitation rate *(first image)* projected on a Cartesian grid *(second image)* and the resulting Cartesian image *(on the right)* obtained after the diffusion process.

5.2.2 Sliding windows

We first remind notations used in this chapter. The state vector $\mathbf{X}$ is composed, at least, of the motion function $\mathbf{w}$ and of an image function I, both defined on the studied space-time domain Ω_T. Depending on the method used for the estimation and if it considers or not structures, $\mathbf{X}$ includes (for Method $\mathcal{M}_2$) or not (for Method $\mathcal{M}_1$) a distance map ϕ for describing regions with high precipitation rates. During the estimation phase, the temporal evolution of $\mathbf{X}$ satisfies an evolution model $\mathbb{M}_e$, described by partial differential equations on $\mathbf{w}$, I (in Methods $\mathcal{M}_1$ and $\mathcal{M}_2$) and on ϕ for Method $\mathcal{M}_2$. The image function I is a passive tracer that has physical properties similar to the image acquisitions I^O. For Method $\mathcal{M}_2$, the distance map ϕ is also considered similar to ϕ^O, which describes the observed position of the regions with high precipitation rates, visible on I^O. The observation vector, containing either I^O for $\mathcal{M}_1$ or both I^O and ϕ^O for $\mathcal{M}_2$, is denoted $\mathbf{Y}$.

Motion estimation relies on three consecutive radar acquisitions that are named $\mathbf{Y}_{e_1}$ to $\mathbf{Y}_{e_3}$. They correspond to a temporal interval with a maximal duration of ten minutes, depending on the

chosen acquisition rate. As extensively described in Chapter 1 for Method $\mathcal{M}_1$, and Chapter 2 for Method $\mathcal{M}_2$, the estimation is based on the assimilation system:

$$\begin{aligned} \frac{\partial \mathbf{X}}{\partial t}(\mathbf{x},t) + \mathbb{M}_e(\mathbf{X})(\mathbf{x},t) &= 0 && (5.3)\\ \mathbf{X}(\mathbf{x},0) &= \mathbf{X}^{(b)}(\mathbf{x}) + \mathcal{E}_B(\mathbf{x}) && (5.4)\\ \mathbb{H}(\mathbf{X},\mathbf{Y})(\mathbf{x},t) &= \mathcal{E}_R(\mathbf{x},t) && (5.5) \end{aligned}$$

and is obtained by minimizing the cost function:

$$\begin{aligned} J(\mathbf{X}(0)) = \frac{1}{2}\int_\Omega \left(\mathbf{X}(0)-\mathbf{X}^{(b)}\right)^T B^{-1}\left(\mathbf{X}(0)-\mathbf{X}^{(b)}\right)\mathrm{d}\mathbf{x} \\ + \frac{1}{2}\int_{\Omega_T} \mathbb{H}(\mathbf{X},\mathbf{Y})^T R^{-1}\mathbb{H}(\mathbf{X},\mathbf{Y})\mathrm{d}\mathbf{x}\mathrm{d}t \\ + \mathcal{R}_1(\mathbf{X}(0)) + \mathcal{R}_2(\mathbf{X}(0)) + \mathcal{R}_3(\mathbf{X}(0)) \quad (5.6) \end{aligned}$$

As image assimilation methods rely on temporal evolution laws, contained in the estimation model $\mathbb{M}_e$, two estimations obtained with the same observations $\mathbf{Y}$, up to one of them, should be close one to each other. A sliding window method is then designed that allows to iterate over time the estimation and forecast components. The principle is to slide in time the temporal interval considered for the estimation. Two consecutive temporal intervals can either overlap or not depending on the operational implementation. However, in our context, the precipitation rates change rapidly in time and the accuracy is of major importance for the monitoring of flash floods. An implementation with non overlapping temporal intervals is then out of question. Moreover, considering overlapping intervals imply to assimilate common observations. This ensures the time coherence of the estimated motion function. The process of sliding windows, as it is implemented in our operational context, is illustrated by Figure 5.6.

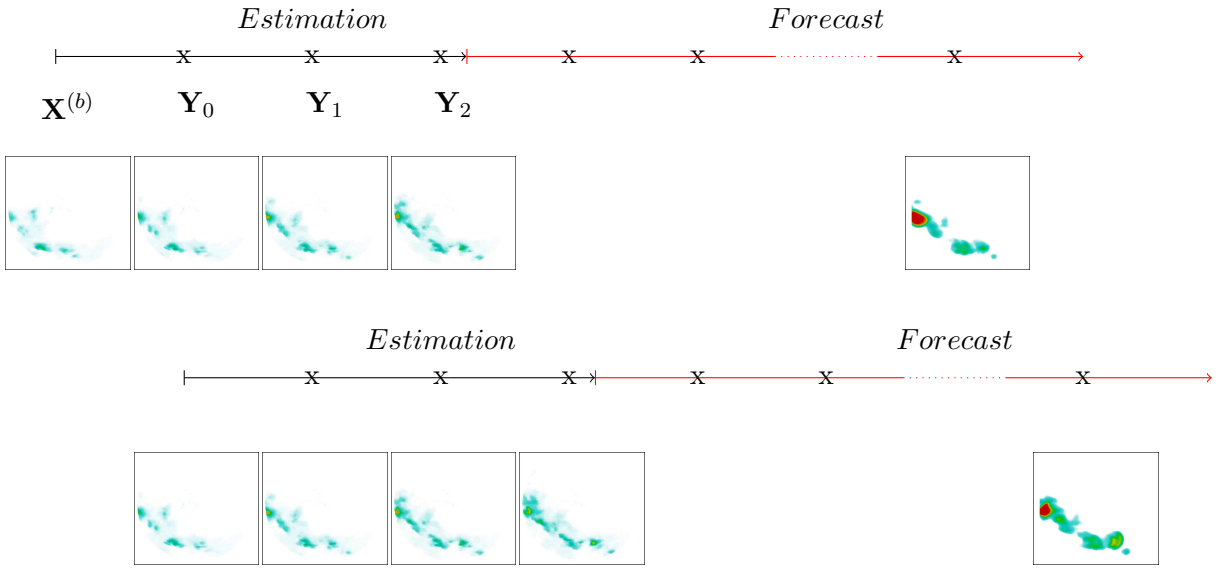

Figure 5.6 – Illustration of the sliding window approach.

For image assimilation in general, and for motion estimation in particular, a background value of the state vector, named $\mathbf{X}^{(b)}$, is needed at initial date of each estimation window. $\mathbf{X}^{(b)}$ represents an empirical estimation of the state vector $\mathbf{X}$. The definition of the background value is different for the first temporal window of the studied rainy event and the next ones. For the first window, the background $I^{(b)}$ of the image function I is the radar acquisition $I^O(0)$ available at initial date of the studied interval. For the motion variable, $\mathbf{w}$, as no empirical estimation is available on the first studied window, its motion background $\mathbf{w}^{(b)}$ is taken as null. For Method $\mathcal{M}_2$ that includes the variable ϕ, the background value $\phi^{(b)}$ is computed on $I^{(b)}$ as explained in Section 2.1 of Chapter 2, with details on computations. On the following windows, information on the previous estimation is used to provide the temporal coherence of the estimation. The background of the image function is the radar acquisition at the beginning of the temporal interval, again denoted $I^O(0)$. The background of motion is defined as the result on the previous window at the same date. This allows the temporal coherence of the estimation and the spatial smoothness of the estimated motion fields. For Method $\mathcal{M}_2$, $\phi^{(b)}$ is again computed on $I^{(b)}$ as explained in Section 2.1 of Chapter 2. The remaining assimilation parameters have been discussed in Subsection 4.2.2 of Chapter 4, when illustrating with results on motion estimation from radar acquisitions.

In order to produce a forecast at a given temporal horizon, one needs an initial motion field $\mathbf{w}$ and an initial image I of the precipitation rate. The operational use of the system may be obtained with two strategies for the forecast. In the first case, a new forecast is produced as soon as a new acquisition is available, using the last estimated motion field. In the second case, the last acquisition is first used (with some of the previous ones) in order to compute a more recent motion field, and a forecast is then produced from this last acquisition and this motion field.
The first strategy allows to obtain a forecast with a shorter delay from the last acquisition date but its motion field relies on older acquisitions.
The second strategy enables to make a forecast relying on the last acquired data, but it adds a delay due to the motion estimation by the image assimilation algorithm.
In an operational system, both strategies should be used alternatively and the update rate of forecasting would be twice the one from the radar acquisition.
However, for sake of simplicity, only the second strategy will now be discussed in the remaining of this chapter. At each acquisition date, a new motion field is estimated from the last acquired data, and a forecast will then be produced from the last radar image and that motion field.

This section has described the acquisition process, the pre-processing applied on the data and the operational framework of sliding windows. For going back to the detailed description of the estimation process, the interested Reader should return to Chapter 1 and Chapter 2. For results of motion estimation on ground-radar data, the Reader should return to Chapter 4. The next section will then directly describe and analyze the results of forecast, depending on the method chosen for the estimation, whether it considers structures (in Method $\mathcal{M}_2$) or not (in Method $\mathcal{M}_1$).

5.3 Forecast

As explained before, the last estimated motion field obtained by the image assimilation Methods ($\mathcal{M}_2$ with structures or $\mathcal{M}_1$ without) is used for integrating in time the evolution equation of the image variable, with the last rain rate acquisition as initial condition. However, the equation on the motion variable, which is included in the estimation model $\mathbb{M}_e$, relies on strong assumptions: time constancy of the motion vector on each pixel trajectory. Therefore a second model, called forecast

model and denoted $\mathbb{M}_f$, is used to forecast the precipitation rates at short temporal horizon. This section describes the forecast model $\mathbb{M}_f$ and discusses the quality of the resulting precipitation rate forecast.

The forecast horizon is at least of 30 minutes, which is approximately three to five times longer than the temporal window used for estimating motion by the image assimilation method. Therefore, the motion field is no more considered as being stationary, as it was the case when designing the estimation model $\mathbb{M}_e$. The heuristic used for the forecast component is that the motion field $\mathbf{w}$ is advected by itself. As previously, the image function I is transported by the optical flow. These two properties are described by the two following equations, defining the forecast model $\mathbb{M}_f$:

$$\frac{\partial \mathbf{w}}{\partial t} + (\mathbf{w}.\nabla)\mathbf{w} = 0 \tag{5.7}$$

$$\frac{\partial I}{\partial t} + \mathbf{w} \cdot \nabla I = 0 \tag{5.8}$$

Equations (5.7) and (5.8) are discretized in time using an Eulerian scheme, which leads to the equation:

$$\mathbf{X}_t = \mathbb{M}_f(\mathbf{X}_{t-1}) \tag{5.9}$$

Two spatial discretization methods have been considered for integrating the forecast model:

- a semi-Lagrangian scheme, which is unconditionally stable, even in case of large motion values, but has the drawback of smoothing the advected quantities,
- an Eulerian scheme, which is more accurate to advect regions where the spatial gradient of the advected function is important, but which is constrained by the CFL condition. Moreover, instabilities occur when integrating the Godunov scheme over a long temporal interval with a small temporal increment.

Considering a forecast horizon of 30 minutes implies either to perform more integration steps of the forecast model or to use larger time steps, compared to the options taken for the 10-minutes estimation interval. As increasing the number of integration steps results in increasing the computational cost, the latter solution has been chosen and new schemes have been investigated for increasing the time step.

Let remind the two types of discretization schemes that are at hand, briefly recall how they work and, finally, design a mixed scheme relying on both of them.

A semi-Lagrangian schemes relies on the computation of a trajectory that is followed back in time to estimate the value of the advected quantity. If integrating the model with a semi-Lagrangian scheme and a large time step dt (for instance several minutes), the trajectory of points and structures can no more be considered as linear, as it was done with the estimation model $\mathbb{M}_e$ in Subsection 1.2.2 of Chapter 1. In order to estimate more accurately the displacements, iterative methods have been developed and a review is given in the paper of Staniforth et al. [29] and detailed furthermore by Temperton et al. in [59] and Hortal et al. in [60]. An iterative procedure works by first computing the displacement α_0, from the motion value $\mathbf{w}$ at pixel $\mathbf{p}$ and time indexes t and $t-1$, thanks to the equation:

$$\alpha_0 = \frac{dt}{2}(3\mathbf{w}(\mathbf{p}, t) - \mathbf{w}(\mathbf{p}, t-1)) \tag{5.10}$$

Then, until convergence is reached, α_{k+1} is computed from α_k and $\mathbf{w}$, using the formula:

$$\alpha_{k+1} = \frac{dt}{2}(2\mathbf{w}(\mathbf{p}+\alpha_k, t) - \mathbf{w}(\mathbf{p}+\alpha_k, t-1) + \mathbf{w}(\mathbf{p}, t)) \tag{5.11}$$

where k denotes the current iteration. Only a few iterations are necessary for ensuring that the difference between the vectors α_{k+1} and α_k becomes negligible, meaning that the displacement value has been correctly estimated. Therefore, the total number of iterations is determined in advance by the user. A convergence criteria, allowing an adaptive number of iterations, can be easily implemented by choosing an appropriate convergence criteria but, for a complete control of the computational time, this has not been implemented in the operational system described in this chapter.

Eulerian schemes are more accurate for advecting functions whose spatial gradients have high values. The motion advection of Equation (5.7) is split into a linear and a non-linear part respectively integrated with a first order Upwind and a Godunov scheme. For a detailed description, the Reader should refer to Subsection 1.2.1 of Chapter 1.

The idea behind the effective implementation of the forecast model is to combine the two types of temporal discretizations, so that the stability of the semi-Lagrangian scheme and the accuracy of the Godunov scheme for computing the advection both benefit the forecast. In order to fulfill the CFL condition, the Godunov scheme is integrated with a one second time step. This temporal integration is only applied on the motion component $\mathbf{w}$ of the state vector $\mathbf{X}$ from 0 to a given date t. Then, at that date t, the semi-Lagrangian model is applied, once, from 0 to t, both for the motion field $\mathbf{w}$ and the image function I. The semi-Lagangian scheme uses a 5 minutes time step. It seems a good compromise between smoothness of the forecast and accuracy of the displacements.

The operational context has been described in Section 5.2. The current section has defined and described the forecast model and its discretization schemes. The full knowledge on the operational system is now available. Section 5.4 will then focus on forecast experiments and on their validation.

5.4 Results and validation

This section will only focus on the forecast results, as Chapter 4 gave an extensive comparison of estimation results using the methods described in Chapter 1 and Chapter 2.

First, several validation approaches are discussed, each of them illustrated with forecast results obtained from an estimation with Method $\mathcal{M}_2$. Then, having define the best approach to validate the results, comparison between the forecasts obtained with the two Methods, $\mathcal{M}_1$ and $\mathcal{M}_2$, will be given.

The quality of the forecasts should be evaluated for at least one hour duration, according to the operational purposes. As no ground truth on motion is available, the only validation of the method comes from the comparison of $I_f(t_i)$ (where f stands for $f$orecast), which is the image function of $\mathbf{X}$ at date t_i of the forecast phase, and $I_f^O(t_i)$, which is the radar acquisition at date t_i. However, their discrepancy comes not only from the inaccuracy of the estimated motion field, but also from the physical processes that provoke appearance or disappearance of clouds on $I_f^O(t_i)$ and of the limitations of the numerical forecast model.

The first result is obtained by the forecast model $\mathbb{M}_f$ described in Subsection 5.3, for the fortieth estimation window (this window is only chosen for illustration purposes). Figure 5.7 shows, from left to right, the comparison between forecasts and acquisitions, at temporal horizons of fifteen

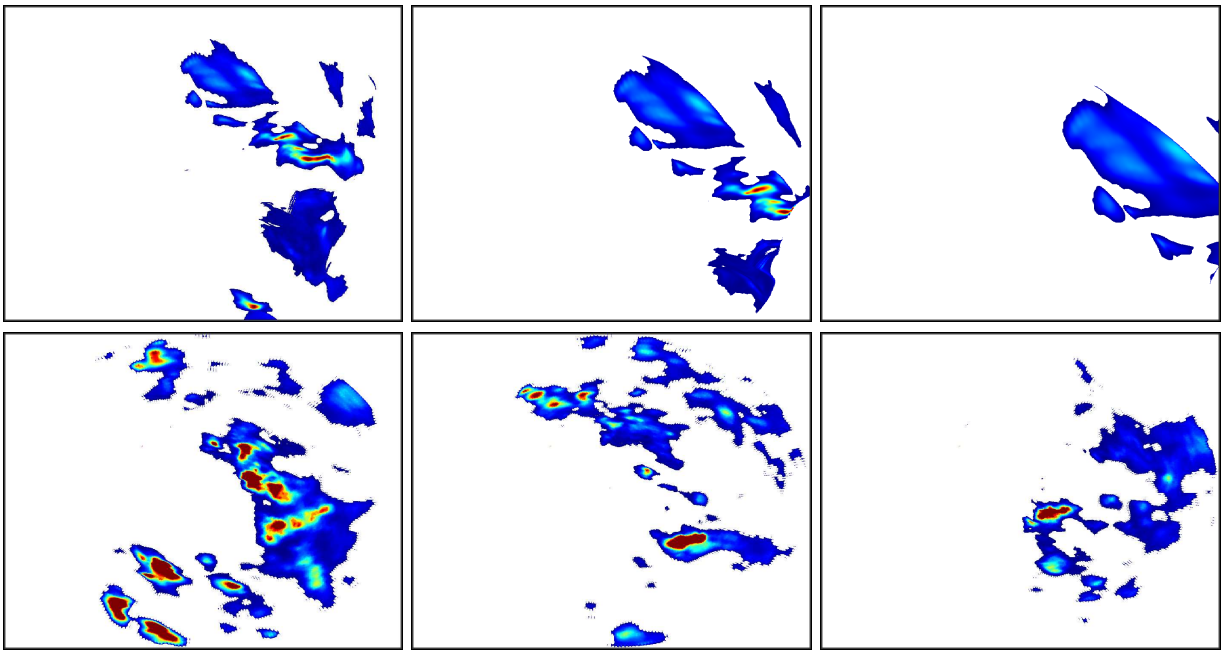

Figure 5.7 – Forecasted precipitation rate images *(top line)*, at horizons of 15, 30 and 60 minutes *(from left to right)*. Corresponding acquisitions *(bottom line)*.

minutes, thirty minutes and one hour. The comparison of couples of corresponding forecast I_f and acquisition I_f^O is difficult. Moreover, from an operational point of view, these comparisons are of little interest. In fact, instantaneous precipitation rates at pixel scale will not be used by the decision making system.

According to the foreseen operational exploitation, it is less important to forecast an exact precipitation rate at a specific date than to accurately forecast the global precipitations quantity occurring on a region over a whole period. For that purpose, the precipitation rate is integrated in time and the average values of the forecasted and acquired images compared. The results of this process is shown on Figure 5.8. The figure displays, on the first and third column, the measured precipitations quantity over 15 and 30 minutes. The second and fourth column are the respective values of the forecast. It can be seen, comparing the regions in green, where precipitations are occurring, that their shapes are quite well forecasted. The main drawback, as visible on the first line of Figure 5.8, is that the forecasting system only propagates the last measured precipitation values and does involve the physical processes responsible of the creation and disappearance of the precipitation cells. All studied sequences start from a situation without rain. Then, they display a storm going through the radar domain. They end when the storm is finished. When applying the forecast process at the beginning of such sequences, the result tends to underestimate the rain quantity. During the storm period, the forecast is quite accurate. At the end of the sequence, the forecast overestimates the precipitation quantity. This drawback is inherent to all the image-based nowcasting methods and is a challenging issue to be solved.

An interesting output of the system is the detection of threshold exceedance over regions, when the cumulative rain quantity is above a given value that requires emergency measures. The percentage of correct forecast at 1 hour of these threshold exceedance events (this means that

the forecast corresponds to a real event) is around 98% with both Method $\mathcal{M}_1$ and Method $\mathcal{M}_2$. However, the high percentage of forecasting of these threshold exceedance events is partly due to the overestimation of the precipitation already noticed on Figure 5.8. In fact, if the system forecasts rain everywhere, no threshold exceedance event will be missed.

Preventive measures should be applied only if the threat is justified and overestimation is an issue. Therefore, the percentage of alerts sent by the system and actually measured is an important criteria. The involvement of a structures representation ϕ in the state vector of Method $\mathcal{M}_2$ has a positive or neutral impact on that criteria. In average, the percentage of detection at 1 hour of real events is close to 68% with $\mathcal{M}_2$ (including the structure variable) and decreases to 62% with $\mathcal{M}_1$ (without structure variable), as displayed in Figure 5.10.

5.5 Conclusion and perspectives

This chapter focused on the development of an operational system for nowcasting precipitations which relies on the estimation methods described in Chapter 1 and Chapter 2 and on ground-radar acquisitions. A full description of the radar acquisitions was given in Section 5.2 as well as a description of the sliding window framework. A forecast model $\mathbb{M}_f$ is required to generate the forecast from the estimation of the system state. The model $\mathbb{M}_f$ has been described in Section 5.3. Last, the system has been tested on 5 sequences of more than 50 acquisitions each, obtained during summer 2014. The forecasts have been compared to the real acquisitions at the same dates. Results are encouraging and a number of perspectives has been defined for improving the system with the company Weather Measures.

First, the estimation model $\mathbb{M}_e$ and the whole estimation component should be rewritten in polar coordinates, according to the acquisition process. This would enable to rely on the effective acquisitions without any pre-processing. In particular, the diffusion process, see 5.2.1, will be suppressed. Second, as done by Ridal et al. [24], the forecast could be obtained by combining results of the forecast model $\mathbb{M}_f$ and those obtained with a numerical weather prediction method. Such an approach has proven its ability to produce short term and accurate forecasts and allows to consider the appearance or disappearance of precipitation cells. This would permit to increase the time horizon and also improve the accuracy of long-term forecasts. Third, the pre-processing of the acquisitions should be revisited in order to diminish the acquisition problems described in Subsection 5.2.1. An improved pre-processing should lead to a better quality forecast. Last, the validation could be done, for instance, by using rain gauges and comparing the forecasted precipitation values to effective rain measures.

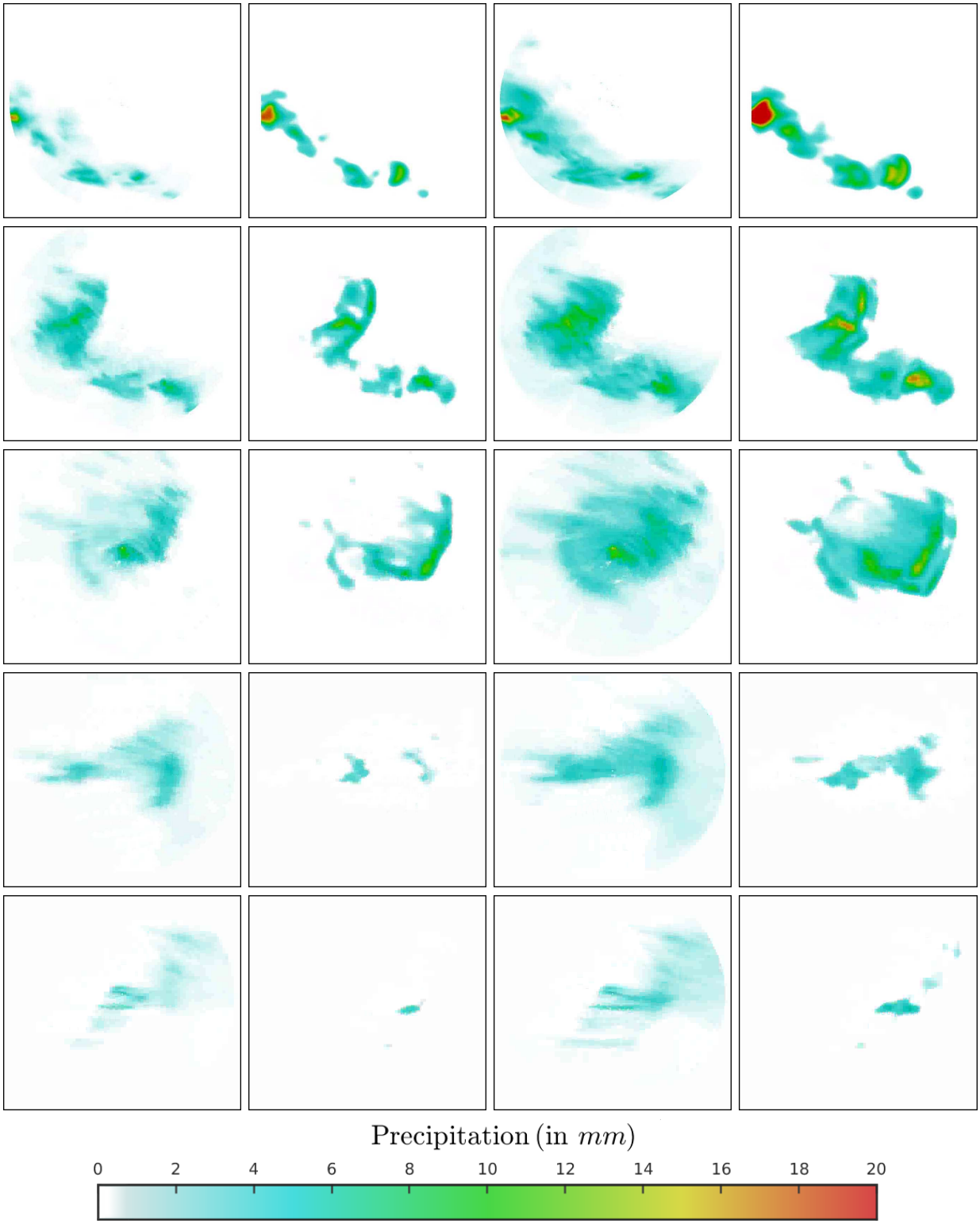

Figure 5.8 – Average of the acquired precipitation quantities on temporal intervals of 15 minutes (*first column*) and 30 minutes *(third column)* and the corresponding forecasted values *(second and fourth column)*. The vertical axis corresponds to different dates of the sliding window approach.

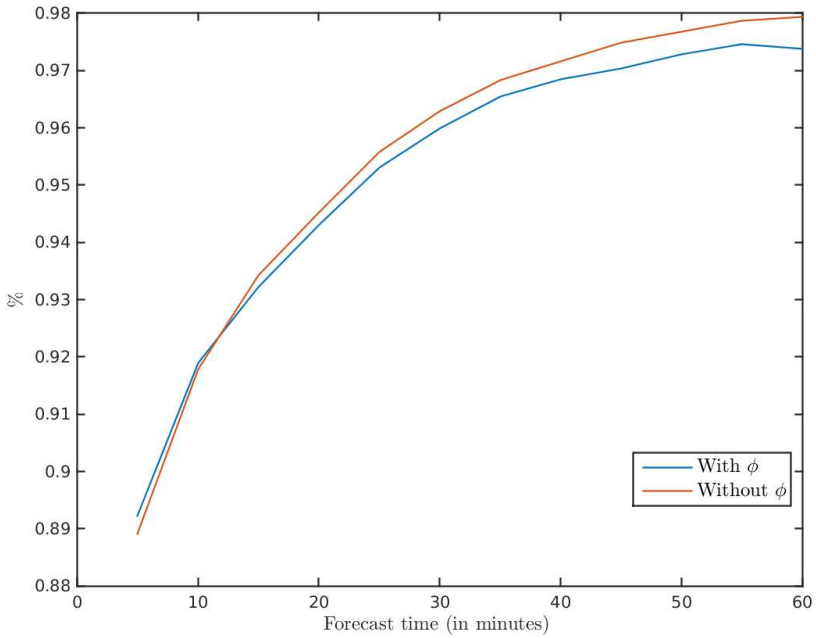

Figure 5.9 – Percentage of measured events that were forecasted. With (blue) and without (red) ϕ.

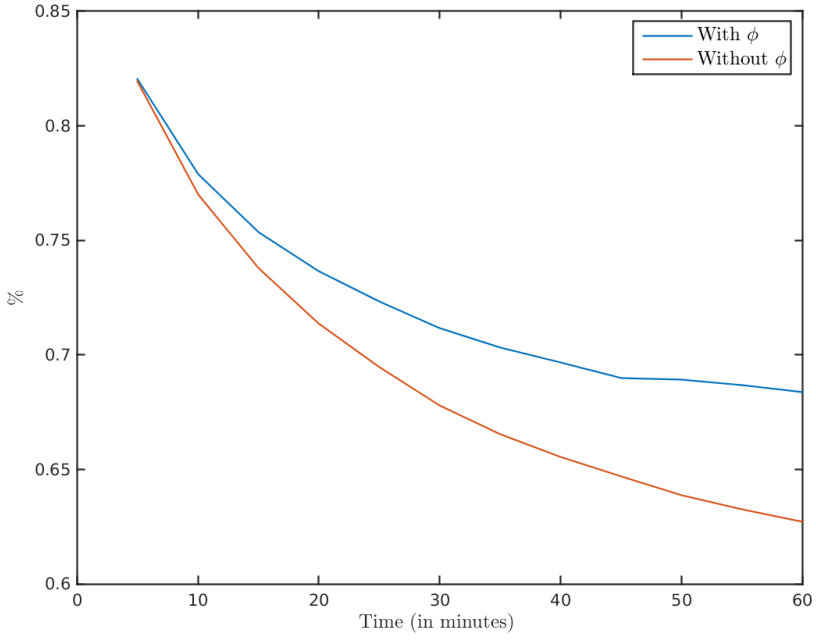

Figure 5.10 – Percentage of alerts for existing events. With (in blue) and without (in red) ϕ.

Part II

An image-based ensemble Kalman filter for motion estimation with structure components

Chapter 6

Motion estimation with an Image-Based Ensemble Kalman Filter

This chapter revisits the use of optical flow methods for estimating motion on discrete image sequences. Based on the concept of Data Assimilation (DA), we design a filter, which relies on the evolution laws of the dynamics underlain in the image sequence. The aim is to retrieve a dense-in-time motion estimation from a sparse-in-time discrete image sequence. 4D-Var data assimilation methods have been successfully used for motion estimation, as described in Part I. However, these approaches present some major restrictions:

- the estimate is obtained from the whole set of images acquired during the studied temporal interval. It corresponds to a smooth compromise between all these observations.
- an adjoint model is required, which needs additional theoretical work and software development,
- processing an image sequence requires a number of forward integration of the model and backward integration of its adjoint during the optimization process. Even if the code is parallelized, the whole process necessitates more computational time than a single integration of the model, as done with a filtering approach.
- they provide an estimate of the motion field on the whole temporal interval, but no uncertainty measure of the result is at hand. That uncertainty value is however mandatory for further interpretation of motion and operational use of results.

The design of filtering methods, such as the Kalman filter, is not affected by the previous limitations. However, the original Kalman filter, presented in [25], suffers from unaffordable computational costs and memory requirements if applied for the large state vector $\mathbf{X}$ involved by images. A filtering method based on the **En**semble **K**alman **F**ilter (**EnKF**) [4] is an alternative that is not affected by any of the previous limitations. EnKF is an adaptation of the Kalman filter that does not use an analytical description of the **P**robability **D**ensity **F**unction (**PDF**), but samples it. To implement the filter, an ensemble of motion fields is constructed at initial date and propagated in time. At each date, an estimate of motion is given by the mean of the members and its uncertainty is described by the spread of this ensemble.

This chapter describes the design of an image-based ensemble Kalman filter, whose components are defined only from image properties, which is used to estimate motion from image sequences.

Section 6.1 defines the mathematical notations and describes the EnKF method, so that the Reader can have a complete knowledge on the subject, before reading the description of how it is adapted for motion estimation from image sequences. Subsection 6.1.2 describes, starting from Bayes' formula, how to derive the Kalman equations and why it is necessary for our application to consider the EnKF approximation.

Section 6.2 describes the main characteristics of this **I**mage-based **EnKF** (named **IEnKF**) and the experimental implementation. A brief summary of the Image Model $\mathbb{IM}$ is given in Subsection 6.2.1. The propagation in time of the ensemble members relies on the following evolution laws: transport of the image brightness by Euler equations and the motion field for the motion function. These heuristics are similar to those that have been presented in Part I. The Reader can go back to Equation (1.11) and Equation (1.19) of Chapter 1 for a more complete description. Subsection 6.2.2 explains how the initial motion ensemble is designed. If considering that the PDF of the uncertainty of motion is described by a zero-mean normal law, its standard deviation at initial date, is computed from a set of motion results on the two first frames, which are obtained by a large number of optical flow methods, with varying parameters values.

The different members of an Ensemble Kalman Filter usually come closer one to each other after each analysis step, applied during temporal integration. This is named the "shrinking of the ensemble". To avoid this drawback, several approaches are at hand. For instance, the deterministic EnKF, presented by Sakov et al. [61], corrects the members after the analysis, so that the covariance matrix computed from the new members gets closer to the one theoretically computed by the Kalman Filter equations. However, applying this correction after the analysis step conducts to an overload of computations. Consequently, this approach has not been used in the document. As explained in Section 6.3, in order to avoid shrinking, observations ensembles [62] are needed if using the EnKF as such. The construction of the observations ensembles is based on characteristics of the image acquisitions and also described in Section 6.3.

EnKF methods suffer from the approximation of the background error covariance matrix by an ensemble of members. This sampling approximation leads to high spurious covariance values in the matrix. Several approaches were designed to counterbalance this effect. Inflation and localization methods, as presented by Anderson et al. [63], Hamill et al. [64] or Oke et al. [65], respectively increase the variance values and decrease the covariance values of the matrix computed on the ensemble members, before applying the analysis step. A precise tuning of the inflation parameter is required to simultaneously get accurate results and keep the spread of the ensemble. Several methods of the literature automatically define the optimal value, as described by Anderson et al. [66]. However, their definition of the inflation parameter may not be directly linked to the image properties and is not suitable for an image-based approach. Alternatively, a localization method, which depends on image properties, is easily defined, as discussed in Section 6.4. It involves: first, the distance between pixels (pixels separated by a high distance are usually independent), second, the similarities between pixels (pixels belonging to the same structures are usually highly correlated), which enable to take into account the information given by the structures displacement during the assimilation process.

6.1 Ensemble Kalman Filter

This section describes, in a first subsection, the mathematical notations used in this chapter, but also in the whole Part II. Some of these notations are specific to sequential data assimilation.

Others are just reminders of the notations used in Part I, and a few of them are slightly modified to render the discussion on ensemble members more comprehensible. A second subsection describes the Kalman filter, starting from Bayes' formula. It also discusses some drawbacks of the Kalman filter and explains the necessity of the **En**semble **K**alman **F**ilter, denoted **EnKF** in the following.

6.1.1 Mathematical Setting

- A point is denoted $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$. It belongs to the continuous image domain Ω.

- The sequence of $\mathrm{N}_O + 1$ images $\left\{ I^O(t), \quad t \in [0, \mathrm{N}_O] \right\}$ is defined on Ω. It has been acquired on a discrete temporal interval $[0, T]$. For more readable equations, $I^O(t)$ will sometimes be denoted I_t^O in the following.

- The data assimilation method requires the definition of a state vector $\mathbf{X}$, whose value at date t is $\mathbf{X}(t) = \begin{pmatrix} \mathbf{w}(t) \\ I(t) \end{pmatrix}$. $\mathbf{w}(t) = \begin{pmatrix} u(t) \\ v(t) \end{pmatrix}$ denotes the motion field and $I(t)$ is a synthetic image function with the same physical properties than the studied image sequence. The state vector $\mathbf{X}(t)$ at date t will also be denoted $\mathbf{X}_t$ (as well as for u_t, v_t, $\mathbf{w}_t$ and I_t).

- If an image acquisition I_t^O is available at date t, it is used to compute the observation vector $\mathbf{Y}_t$.

- The aim of data assimilation is to get an estimate, or analysis, $\mathbf{X}_t^{(a)}$ of the true state vector $\mathbf{X}_t^{(r)}$ (r stands for $r$eference) from the so-called background state vector $\mathbf{X}_t^{(b)}$ and, if available, the observation vector $\mathbf{Y}_t$.

- Ensemble methods rely on a number N_m of members that evolve simultaneously in time. $\mathbf{X}_t^i$ denotes the state vector at date t of the i^{th} member of the ensemble. If an observations ensemble is constructed from images, $\mathbf{Y}_t^i$ denotes the i^{th} member of this observation ensemble at date t.

- $\overline{\quad .\quad}$ denotes the mean over the ensemble members. Let denote N_m the number of members in the observations ensemble and the state vectors ensemble. At date t, the mean $\mathbf{X}_t$, also denoted $\overline{\mathbf{X}_t^i}$, of the state vectors $\mathbf{X}_t^i$ is defined by:

$$\mathbf{X}_t = \overline{\mathbf{X}_t^i} = \frac{1}{\mathrm{N}_m} \sum_{i=1}^{\mathrm{N}_m} \mathbf{X}_t^i \tag{6.1}$$

  Similarly, at date t, the mean $\mathbf{Y}_t$, also denoted $\overline{\mathbf{Y}_t^i}$, of the observation vectors $\mathbf{Y}_t^i$ is defined by:

$$\mathbf{Y}_t = \overline{\mathbf{Y}_t^i} = \frac{1}{\mathrm{N}_m} \sum_{i=1}^{\mathrm{N}_m} \mathbf{Y}_t^i \tag{6.2}$$

- Having defined the state vectors ensemble composed of the $\mathbf{X}_t^i$, and the observations ensemble composed of the $\mathbf{Y}_t^i$, let define their respective perturbations ensemble, $\mathcal{X}_\mathrm{t}$ and $\mathcal{Y}_\mathrm{t}$, by the

matrices:

$$\mathcal{X}_{\mathrm{t}} = \left[\left(\mathbf{X}_t^1 - \overline{\mathbf{X}_t^i}\right) \quad \dots \quad \left(\mathbf{X}_t^i - \overline{\mathbf{X}_t^i}\right) \quad \dots \quad \left(\mathbf{X}_t^{\mathrm{N}_m} - \overline{\mathbf{X}_t^i}\right)\right] \tag{6.3}$$

$$\mathcal{Y}_{\mathrm{t}} = \left[\left(\mathbf{Y}_t^1 - \overline{\mathbf{Y}_t^i}\right) \quad \dots \quad \left(\mathbf{Y}_t^i - \overline{\mathbf{Y}_t^i}\right) \quad \dots \quad \left(\mathbf{Y}_t^{\mathrm{N}_m} - \overline{\mathbf{Y}_t^i}\right)\right] \tag{6.4}$$

- Some error terms are needed to describe the method. $\mathcal{E}_{R_t}$ will denote a centered Gaussian noise associated to the covariance matrix R_t. The error will also be denoted:

$$\mathcal{E}_R \sim \mathcal{N}\left(0, R_t\right) \tag{6.5}$$

- Let denote $p\left(\mathbf{X}_t\right)$ the probability density function of the random variable $\mathbf{X}_t$ and $p\left(\mathbf{X}_t|\mathbf{Y}_t\right)$ the conditional probability density function of the random variable $\mathbf{X}_t$ given $\mathbf{Y}_t$.
- Let denote $\mathrm{N}_{\mathbf{X}}$ the size of the state vector $\mathbf{X}_t$ and $\mathrm{N}_{\mathbf{Y}}$ the size of the observation vector $\mathbf{Y}_t$. These numbers will be needed to discuss implementation issues.

Having defined these notations it is possible to proceed to the description of the algorithm.

6.1.2 Algorithm

The idea behind data assimilation is to use the output $\mathbf{X}^{(b)}$ of a model and some measurements $\mathbf{Y}$, in order to produce an estimation $\mathbf{X}^{(a)}$ of the real state $\mathbf{X}^{(r)}$ of the system. However, the proposed framework should also estimate the uncertainty on the analysis. Therefore, let first define the errors and associated PDF for the concerned variables. As for the 4D-Var algorithm described in Part I, the errors are supposed Gaussian and zero-mean. The background error $\mathcal{E}_{B^{(b)}}$ represents the a priori error on the background state vector $\mathbf{X}^{(b)}$, mathematically written as:

$$\mathbf{X} = \mathbf{X}^{(b)} + \mathcal{E}_{B^{(b)}} \tag{6.6}$$

It comes that the law of $\mathbf{X}$ is defined by:

$$\mathbf{X} \sim \mathcal{N}\left(\mathbf{X}^{(b)}, B^{(b)}\right) \tag{6.7}$$

with $B^{(b)}$ the covariance matrix associated to $\mathcal{E}_{B^{(b)}}$.

The measurement error $\mathcal{E}_R$ represents the error done when observing the variable $\mathbf{X}$, leading to the so-called measurement model:

$$\mathbf{Y} = \mathbb{H}\mathbf{X} + \mathcal{E}_R \tag{6.8}$$

Therefore the law of $\mathbf{Y}$ given $\mathbf{X}$ is defined by:

$$\mathbf{Y}|\mathbf{X} \sim \mathcal{N}\left(\mathbb{H}\mathbf{X}, R\right) \tag{6.9}$$

with R being the covariance matrix associated to $\mathcal{E}_R$.

The goal is to compute the conditional probability density function $p\left(\mathbf{X}|\mathbf{Y}\right)$ of $\mathbf{X}$ given $\mathbf{Y}$, which will give the most probable outcome $\mathbf{X}^{(a)}$ and an estimation of the uncertainty. The Bayes' formula gives a perfect framework to achieve this:

$$p\left(\mathbf{X}|\mathbf{Y}\right) = \frac{p\left(\mathbf{X}\right) p\left(\mathbf{Y}|\mathbf{X}\right)}{p\left(\mathbf{Y}\right)} \tag{6.10}$$

$p(\mathbf{X})$ and $p(\mathbf{Y}|\mathbf{X})$ are respectively deduced from Equation (6.7) and Equation (6.9). Let remark that, in the conditional formulation of Bayes' equation, Equation (6.10), the integration of $p(\mathbf{X}|\mathbf{Y})$ from $-\infty$ to ∞ is equal to one. Therefore, knowing $p(\mathbf{X})$ and $p(\mathbf{Y}|\mathbf{X})$, the value $p(\mathbf{Y})$ is only considered as a normalizing constant and is no more considered in the description of the estimation method. Let remind that $\mathbf{X} \sim \mathscr{N}\left(\mathbf{X}^{(b)}, B^{(b)}\right)$. This leads, by definition, to the equation of the PDF of $\mathbf{X}$:

$$p(\mathbf{X}) = \frac{1}{(2\pi)^{\frac{\mathrm{N}_\mathbf{X}}{2}}|B^{(b)}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{X}-\mathbf{X}^{(b)})^T {B^{(b)}}^{-1}(\mathbf{X}-\mathbf{X}^{(b)})\right] \tag{6.11}$$

and $\mathbf{Y} \sim \mathscr{N}(\mathbb{H}\mathbf{X}, R)$ leads to:

$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{(2\pi)^{\frac{\mathrm{N}_\mathbf{Y}}{2}}|R|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{Y}-\mathbb{H}\mathbf{X})^T R^{-1}(\mathbf{Y}-\mathbb{H}\mathbf{X})\right] \tag{6.12}$$

Therefore, by doing the product of Equation (6.11) and Equation (6.12), it comes that the conditional PDF $p(\mathbf{X}|\mathbf{Y})$ of $\mathbf{X}$ given $\mathbf{Y}$ is proportional to:

$$\exp\left[-\frac{1}{2}\left((\mathbf{X}-\mathbf{X}^{(b)})^T {B^{(b)}}^{-1}(\mathbf{X}-\mathbf{X}^{(b)}) + (\mathbf{Y}-\mathbb{H}\mathbf{X})^T R^{-1}(\mathbf{Y}-\mathbb{H}\mathbf{X})\right)\right] \tag{6.13}$$

To proceed with the calculations of $p(\mathbf{X}|\mathbf{Y})$, let denote J the exponent:

$$J = (\mathbf{X}-\mathbf{X}^{(b)})^T {B^{(b)}}^{-1}(\mathbf{X}-\mathbf{X}^{(b)}) + (\mathbf{Y}-\mathbb{H}\mathbf{X})^T R^{-1}(\mathbf{Y}-\mathbb{H}\mathbf{X}) \tag{6.14}$$

Let rewrite J as a sum of scalar products:

$$J = \left\langle \mathbf{X}-\mathbf{X}^{(b)} \ , \ {B^{(b)}}^{-1}(\mathbf{X}-\mathbf{X}^{(b)}) \right\rangle + \left\langle \mathbf{Y}-\mathbb{H}\mathbf{X} \ , \ R^{-1}(\mathbf{Y}-\mathbb{H}\mathbf{X}) \right\rangle \tag{6.15}$$

Using the bilinearity of the scalar product, Equation (6.15) is rewritten as:

$$\begin{aligned} J = &\left\langle \mathbf{X} \ , \ {B^{(b)}}^{-1}\mathbf{X} \right\rangle - \left\langle \mathbf{X}^{(b)} \ , \ {B^{(b)}}^{-1}\mathbf{X} \right\rangle - \left\langle \mathbf{X} \ , \ {B^{(b)}}^{-1}\mathbf{X}^{(b)} \right\rangle + \left\langle \mathbf{X}^{(b)} \ , \ {B^{(b)}}^{-1}\mathbf{X}^{(b)} \right\rangle \\ &- \left\langle \mathbf{Y} \ , \ R^{-1}\mathbb{H}\mathbf{X} \right\rangle - \left\langle \mathbb{H}\mathbf{X} \ , \ R^{-1}\mathbf{Y} \right\rangle + \left\langle \mathbb{H}\mathbf{X} \ , \ R^{-1}\mathbb{H}\mathbf{X} \right\rangle + \left\langle \mathbf{Y} \ , \ R^{-1}\mathbf{Y} \right\rangle \end{aligned} \tag{6.16}$$

The goal being to write the probability density function of $\mathbf{X}$ given $\mathbf{Y}$, the terms $\langle \mathbf{Y} \ , \ R^{-1}\mathbf{Y} \rangle$ and $\left\langle \mathbf{X}^{(b)} \ , \ {B^{(b)}}^{-1}\mathbf{X}^{(b)} \right\rangle$ are considered constant and, therefore, hidden into a constant C_1. Considering $\mathbb{H}$ as real and linear, its adjoint is equal to its transpose. By factoring Equation (6.16), is comes:

$$\begin{aligned} J = &\left\langle \mathbf{X} \ , \ \left({B^{(b)}}^{-1} + \mathbb{H}^T R^{-1}\mathbb{H}\right)\mathbf{X} \right\rangle \\ &- \left\langle \mathbb{H}^T R^{-1}\mathbf{Y} + {B^{(b)}}^{-1}\mathbf{X}^{(b)} \ , \ \mathbf{X} \right\rangle - \left\langle \mathbf{X} \ , \ \mathbb{H}^T R^{-1}\mathbf{Y} + {B^{(b)}}^{-1}\mathbf{X}^{(b)} \right\rangle + C_1 \end{aligned} \tag{6.17}$$

Let denote ${B^{(a)}}^{-1}$ the matrix:

$${B^{(a)}}^{-1} = {B^{(b)}}^{-1} + \mathbb{H}^T R^{-1}\mathbb{H} \tag{6.18}$$

and $\mathbf{X}^{(a)}$:

$$\mathbf{X}^{(a)} = B^{(a)}\left(\mathbb{H}^T R^{-1}\mathbf{Y} + {B^{(b)}}^{-1}\mathbf{X}^{(b)}\right) \tag{6.19}$$

We use these notations to rewrite Equation (6.17). This leads to:

$$J = \left\langle \mathbf{X} - \mathbf{X}^{(a)} \ , \ {B^{(a)}}^{-1} \left(\mathbf{X} - \mathbf{X}^{(a)} \right) \right\rangle + C_2 \tag{6.20}$$

In that Equation (6.20), C_2 is equal to the sum of C_1 and of an additional term not depending of $\mathbf{X}$, which has therefore no impact on the rest of the discussion. Equation (6.20) ends the mathematical computation of the conditional probability density function $p\left(\mathbf{X}|\mathbf{Y}\right)$ of $\mathbf{X}$ given $\mathbf{Y}$. Let recall, that J is the exponent of the exponential in Equation (6.13):

$$p\left(\mathbf{X}|\mathbf{Y}\right) \propto \exp\left(\left\langle \mathbf{X} - \mathbf{X}^{(a)} \ , \ {B^{(a)}}^{-1} \left(\mathbf{X} - \mathbf{X}^{(a)} \right) \right\rangle \right) \tag{6.21}$$

which leads to:

$$\mathbf{X}|\mathbf{Y} \sim \mathcal{N}\left(\mathbf{X}^{(a)}, B^{(a)} \right) \tag{6.22}$$

where the covariance matrix $B^{(a)}$ is given by:

$$B^{(a)} = \left({B^{(b)}}^{-1} + \mathbb{H}^T R^{-1} \mathbb{H} \right)^{-1} \tag{6.23}$$

Therefore, the most probable outcome of $\mathbf{X}$, given the observation $\mathbf{Y}$, is $\mathbf{X}^{(a)}$ and its associated uncertainty is given by the covariance matrix $B^{(a)}$. It is possible to rewrite $B^{(a)}$ by using the Shermann-Morison-Woodburry formula:

$$(A + UDV)^{-1} = A^{-1} - A^{-1}U \left(D^{-1} + VA^{-1}U \right)^{-1} VA^{-1} \tag{6.24}$$

Using the equality given by this formula in Equation (6.23), it comes:

$$B^{(a)} = B^{(b)} - B^{(b)} \mathbb{H}^T \left(\mathbb{H} B^{(b)} \mathbb{H}^T + R \right)^{-1} \mathbb{H} B^{(b)} \tag{6.25}$$

Let reformulate Equation (6.19) by using Equation (6.25). This leads to the well known Kalman equations:

$$\mathbf{X}^{(a)} = \mathbf{X}^{(b)} + B^{(b)} \mathbb{H}^T \left(\mathbb{H} B^{(b)} \mathbb{H}^T + R \right)^{-1} \left(\mathbf{Y} - \mathbb{H} \mathbf{X}^{(b)} \right) \tag{6.26}$$

$$B^{(a)} = B^{(b)} - B^{(b)} \mathbb{H}^T \left(\mathbb{H} B^{(b)} \mathbb{H}^T + R \right)^{-1} \mathbb{H} B^{(b)} \tag{6.27}$$

The previous mathematical computation of the conditional probability density function $p\left(\mathbf{X}|\mathbf{Y}\right)$, from the estimations of the errors on $\mathbf{X}^{(b)}$ and $\mathbf{Y}$, has been done without any specification of the temporal date t. However, the Kalman filter, as described by Kalman in [25], starts with an initial condition $\mathbf{X}_0^{(b)}$ which evolves in time until an acquisition date t is reached. Having computed a background value $\mathbf{X}_t^{(b)}$ at date t, the observational data $\mathbf{Y}_t$ are used to compute the analysis value $\mathbf{X}_t^{(a)}$. As a result, it provides an approximation $\mathbf{X}_t^{(a)}$ of the true state $\mathbf{X}_t^{(r)}$ at that date, associated with an uncertainty value $B_t^{(a)}$ of this estimation. The process may be summarized by the following description.

Given the following elements:

A. a background value $\mathbf{X}_0^{(b)}$ is given at initial date. This value comes from a previously studied temporal interval or from heuristics on the system. The uncertainty on this background value is described by a probability density function, which is supposed to be a zero-mean normal law with the covariance matrix $B_0^{(b)}$.

B. an observation $\mathbf{Y}_t$, at a date t for which an image is available (we assume a sparse in time image sequence as usual for our applications). The observation vector $\mathbf{Y}_t$ is computed from the image. Its uncertainty is described by a zero-mean normal law of covariance matrix R_t.

C. the time integration of the state vector from date $t-1$ to t is performed by a linear model $\mathbb{M}$:

$$\mathbf{X}_t = \mathbb{M}(\mathbf{X}_{t-1}) \tag{6.28}$$

D. at date t, the state vector $\mathbf{X}_t^{(b)}$, resulting from the previous integration of the model, and the observation vector $\mathbf{Y}_t$ are compared, thanks to the observation operator $\mathbb{H}$. This operator is defined so that the value $\mathbb{H}(\mathbf{X}_t)$ belongs to the observational space.

The equations of the Kalman filter are the following:

1. At date t, the background value $\mathbf{X}_t^{(b)}$ is obtained from the temporal integration of the estimation at date $t-1$, also named analysis $\mathbf{X}_{t-1}^{(a)}$, by applying:

$$\mathbf{X}_t^{(b)} = \mathbb{M}\mathbf{X}_{t-1}^{(a)} \tag{6.29}$$

The propagation in time of the uncertainty covariance matrix $B_t^{(b)}$ satisfies:

$$B_t^{(b)} = \mathbb{M}B_{t-1}^{(a)}\mathbb{M}^T \tag{6.30}$$

2. If no observation is available at date t, the estimation $\mathbf{X}_t^{(a)}$ and its uncertainty $B_t^{(a)}$ are taken equal to that of the background.

3. If an observation vector $\mathbf{Y}_t$ is available at t, then the state vector analysis is computed by the analysis Equation (6.26):

$$\mathbf{X}_t^{(a)} = \mathbf{X}_t^{(b)} + K\left(\mathbf{Y}_t - \mathbb{H}\mathbf{X}_t^{(b)}\right) \tag{6.31}$$

where K is called the Kalman gain and is defined by:

$$K = B_t^{(b)}\mathbb{H}^T\left(\mathbb{H}B_t^{(b)}\mathbb{H}^T + R_t\right)^{-1} \tag{6.32}$$

It should be noted that $\mathbb{H}$ denotes both the linear observation operator and its associated matrix. The uncertainty covariance matrix of the analysis verifies Equation (6.27), which, using the Kalman gain Equation (6.32), is rewritten as:

$$B_t^{(a)} = B_t^{(b)} - K\mathbb{H}B_t^{(b)} \tag{6.33}$$

Having summarized the Kalman filter equations, two major issues have now to be discussed.

- First, the time propagation of the background covariance matrix $B_t^{(b)}$ by the linear model $\mathbb{M}$, written in Equation (6.30), leads to prohibitive computational requirements for large-sized state vectors. For instance, considering an image of 512×512 pixels, the storage of $B_t^{(b)}$ needs more than 500 gigabytes. Moreover, the propagation in time of $B_t^{(b)}$, using Equation (6.30), requires to apply products of matrices and leads to a computational complexity on the order of 10^{11} operations for an image of size 512×512.

- Second, if $\mathbb{M}$ is non linear, we have to consider its approximation by its tangent linear model. The tangent linear is then used for the temporal propagation of the uncertainty in Equation (6.30). However, the resulting value $B_t^{(b)}$ is only an approximation of the true value, due to the approximation of the model. In this document, the Image Model $\mathbb{M}$ is expressing the Lagrangian constancy of velocity and includes non linear terms. The temporal propagation of the uncertainty covariance matrix is then strongly affected by the approximation of the model by its tangent.

These two issues of computational burden and approximation are the reason for exploring the use of an Ensemble Kalman filter, first described by Evensen in [67]. An ensemble of background state vectors $\mathbf{X}_t^{(b),i}$ is defined at each date t. It samples the uncertainty covariance matrix $B_t^{(b)}$. Let denote $\mathbf{X}_t^{(b)} = \overline{\mathbf{X}_t^{(b),i}}$ the mean over the state vectors ensemble, defined by Equation (6.1). An approximation of the covariance matrix is then obtained from the ensemble using the equation:

$$B_t^{(b)} \approx \overline{(\mathbf{X}_t^{(b),i} - \mathbf{X}_t^{(b)})(\mathbf{X}_t^{(b),i} - \mathbf{X}_t^{(b)})^T} \tag{6.34}$$

As $B_t^{(b)}$ is approximated by an ensemble of state vectors $\mathbf{X}_t^{(b),i}$ sampling the probability density function of $\mathbf{X}$, there is no more need to store the whole $\mathrm{N}_{\mathbf{X}}^2$ matrix but only a $\mathrm{N}_m\mathrm{N}_{\mathbf{X}}$ matrix containing every N_m member. As $\mathrm{N}_{\mathbf{X}}$ is usually huge (its size is for instance equal to 10^7 for a 512×512 pixels image) compared to N_m (a few hundreds in the best case), the storage is drastically reduced. Moreover, the propagation in time of $B_t^{(b)}$ does not rely anymore on Equation (6.30), but on the propagation in time of each member $\mathbf{X}_t^{(b),i}$ by the evolution Equation (6.29). This last equation only requires $\mathrm{N}_m\mathrm{N}_{\mathbf{X}}^2$ operations, to be compared with the $\mathrm{N}_{\mathbf{X}}^3$ operations required by Equation (6.30).

Initialized at date 0, the ensemble is propagated in time by integrating each member with the model $\mathbb{M}$, as follow:

$$\mathbf{X}_t^{(b),i} = \mathbb{M}\mathbf{X}_{t-1}^{(a),i} \tag{6.35}$$

If no observation is available at date t, the estimation $\mathbf{X}_t^{(a),i}$ is set equal to $\mathbf{X}_t^{(b),i}$ for each member, and the uncertainty, if needed, is approximated using Equation (6.34).
If an observation $\mathbf{Y}_t$ is available at date t, an analysis is computed for each member i by:

$$\mathbf{X}_t^{(a),i} = \mathbf{X}_t^{(b),i} + K(\mathbf{Y}_t - \mathbb{H}\mathbf{X}_t^{(b),i}) \tag{6.36}$$

where:

$$K = B_t^{(b)}\mathbb{H}^T\left(\mathbb{H}B_t^{(b)}\mathbb{H}^T + R_t\right)^{-1} \tag{6.37}$$

The final estimation at date t is defined as the mean over the ensemble:

$$\mathbf{X}_t^{(a)} = \overline{\mathbf{X}_t^{(a),i}} = \frac{1}{\mathrm{N}_m}\sum_{i=1}^{\mathrm{N}_m}\mathbf{X}_t^{(a),i} \tag{6.38}$$

and its uncertainty is approximated by replacing $^{(b)}$ by $^{(a)}$ in Equation (6.34):

$$B_t^{(a)} \approx \overline{(\mathbf{X}_t^{(a),i} - \mathbf{X}_t^{(a)})(\mathbf{X}_t^{(a),i} - \mathbf{X}_t^{(a)})^T} \tag{6.39}$$

It should be noted that all members are involved in the computation of $B_t^{(b)}$, in Equation (6.34), and therefore in the computation of the Kalman gain by Equation (6.37). Consequently, all members impact the estimation of each value $\mathbf{X}_t^{(a),i}$ in Equation (6.36).

The ensemble Kalman filter, as described by Equations (6.35, 6.36, 6.37, 6.38, 6.39), is highly parallelizable. This comes from the fact that the temporal integration of each member is independent from the others.
However, in order to compute the analysis, all the members need to be gathered before any calculation can be done. The ensemble Kalman filter necessitates to consider an observations ensemble, as it will be explained in Section 6.3. Such an observations ensemble allows an effective implementation of the EnKF analysis step, which relies on a singular value decomposition of $\mathbb{H}^T B_t^{(b)} \mathbb{H} + R_t$, as described by Evensen in [68]. Details of this implementation will be given in Section 6.2.
The independent temporal integration of the members and the efficient implementation of the analysis step make the EnKF highly suitable for operational considerations.

This section has first presented the so-called Kalman filter. Starting from the Bayes' formula and assuming independent Gaussian probability density functions for the prior state vector $\mathbf{X}_t^{(b)}$ and the observation vector $\mathbf{Y}_t$, the filter computes the optimal estimation, denoted $\mathbf{X}_t^{(a)}$, according to this observation vector. Some drawbacks of the Kalman filter have been highlighted, which render impossible its operational implementation for state vectors of large size. We concluded on the interest of the ensemble Kalman filter. The theoretical formulation of this filter has been given. In order to use the EnKF to estimate motion from image sequences, the next section will describe its two main components: the Image Model $\mathbb{M}$ and the initial ensemble construction. However, the ensemble Kalman filter suffers from the sampling estimation of the background covariance matrix $B_t^{(b)}$ through Equation (6.34). This leads to two different problems: the shrinking of the ensemble at the analysis computation and the appearance of spurious covariances in $B_t^{(b)}$. First, the shrinking of the ensemble during the computation of the analysis leads to the impossibility to continue the filtering and to an underestimation of the uncertainty. Second, the appearance of spurious covariances in the background error covariance matrix makes the estimation inaccurate. To avoid the shrinking problem an ensemble of observations is needed, as demonstrated in Section 6.3, where its construction is also described. For limiting the appearance of spurious covariances, a so-called localization method is mandatory and discussed in Section 6.4.

6.2 Implementation of an Ensemble Kalman Filter for Motion Estimation

This section describes the core of this chapter, the design of an ensemble Kalman filter for estimating motion from an image sequence. It is named IEnKF for Image-base Ensemble Kalman Filter. Relying on the EnKF equations, see Equations (6.35, 6.36, 6.37, 6.38, 6.39), the assimilation of the observation $\mathbf{Y}_t$ allows to estimate the state $\mathbf{X}_t^{(a)}$ of a system according to the a priori knowledge $\mathbf{X}_t^{(b)}$.

Two major components of the system, the Image Model $\mathbb{M}$ and the initial ensemble, have to be defined according to the considered problem. Subsections 6.2.1 and 6.2.2 will respectively describe these two issues.

6.2.1 Evolution Model

As previously explained, see Equation (6.35), the ensemble members are integrated in time by the Image Model $\mathbb{M}$ obtained by spatial and temporal discretization of the continuous evolution model, also named $\mathbb{M}$ for simplifying notations.

The heuristics for the evolution of motion and images are identical to those presented in Part I. They correspond to the following equations:

- Lagrangian constancy of velocity $\mathbf{w}$:

$$\frac{d\mathbf{w}}{dt} = \frac{\partial \mathbf{w}}{\partial t} + (\mathbf{w} \cdot \nabla)\mathbf{w} = 0 \tag{6.40}$$

- Transport of the image brightness by the motion field. This is used both for the image acquisitions and for the synthetic image function I included in the state vector:

$$\frac{\partial I}{\partial t} + \mathbf{w} \cdot \nabla I = 0 \tag{6.41}$$

The state vector is defined by $\mathbf{X} = \begin{pmatrix} \mathbf{w} \\ I \end{pmatrix}$. Equations (6.40) and (6.41) are then summarized by the following equation:

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}(t)) = 0 \tag{6.42}$$

that defines the continuous evolution model $\mathbb{M}$.

Temporal discretization of Equation (6.42) with an explicit Euler scheme leads to the discrete equation:

$$\mathbf{X}_t = \mathbb{M}(\mathbf{X}_{t-1}) \tag{6.43}$$

that is applied for propagating in time each member of the ensemble. For the discretization schemes associated to Equation (6.43), the Reader should refer to Chapter 1 of this document, where an extensive description has been given.

6.2.2 Image-based Ensemble Construction

The result of motion estimation with an Image-based Ensemble Kalman filter highly depends on the design of the initial ensemble at the beginning of the studied temporal interval. This ensemble should span a vector space that contains the truth and properly sample the uncertainties. In the literature on EnKF, the initial ensemble is usually obtained by adding some perturbations to an initial guess with a Monte Carlo method. In the case of motion estimation, this subsection describes an innovative alternative, where the ensemble is directly designed from the image observations.

Each member $\mathbf{X}_t^i$ is composed of a motion field $\mathbf{w}_t^i$ and an image I_t^i. However, there is no need to construct an ensemble of images, as the image component is only used as a display of the underlying dynamics. Consequently, at initial date, the various members include the same image, which is chosen as the first acquisition I_0^O, but various motion fields $\mathbf{w}_0^i$. It should be noted that at dates different from 0, the image fields of the members are different, as the same initial image has been integrated by different motion fields.

The motion component, $\mathbf{w}_0^i$, is obtained by adding a Gaussian vector field to a given estimate of the motion field on the first two frames, I_0^O and I_1^O, of the acquisition sequence. The standard

deviation of the normal law that is considered for motion is computed from a set of motion results obtained by a large number of optical flow methods of the literature, with varying parameters values. This standard deviation describes the uncertainty on the chosen motion value at initial date.

There are lots of methods available in the literature of image processing and computer vision in order to estimate motion from two image acquisitions. The Reader may for instance consider the survey from Sun et al. [11], the one from Baker et al. [13] or the more recent one from Fortun et al. [12]. The motion estimation approaches differ in their formulation of the optical flow equation (that rules the temporal evolution of image brightness), in the spatial smoothness assumption, and in the implementation tools: coarse-to-fine estimation, texture decomposition, median filtering, ... For each approach, various results are obtained according to the chosen parameters values. One method may be the best for one part of the temporal sequence, while another one succeeds in the remaining. The same observation is valid in space for different locations of the image domain. This remark conducted to the design of new optical flow methods, in which the data term is varying in the space-time domain, according to image properties. This was, for instance, presented by Aodha et al. in their paper [69]. Being in the same spirit, we came to the conclusion that generating a set of motion fields from a large number of codes, varying the formulations and parametrizations, is the best way to ensure spanning a large vectorial space, which should include the true motion, and to assess the uncertainty on the estimation.

The optical flow algorithms, applied in the chapter, are variational methods that rely on the brightness constancy hypothesis, used by Horn et al. [5]. They estimate motion by minimizing a cost function J, which includes, at least, a data term $f(\cdot)$ and a weighted regularization term $\mathcal{R}(\cdot)$:

$$J(\mathbf{w}) = f\left(I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{w}(\mathbf{x}))\right) + \lambda\left(\mathcal{R}(\mathbf{w})\right) \tag{6.44}$$

Modifying the value of the parameter λ allows the user to choose the smoothness properties of the solution. A small value of λ conducts to a result strongly relying on the data and containing outliers linked to the acquisition noise, whereas a higher one produces a smooth solution, as shown on Figure 6.1. Three types of norms are implemented to compute the value of cost function:

- the quadratic norm, as used by Horn and Schunk [5]:

$$f\left(\mathbf{x}\right) = \mathbf{x}^2 \tag{6.45}$$

- the Charbonnier norm, used by Brox et al. in [70]:

$$f\left(\mathbf{x}\right) = \sqrt{\mathbf{x}^2 + \mathcal{E}} \tag{6.46}$$

- the Lorentzian norm, used by Black and Anandan in [71]:

$$f\left(\mathbf{x}\right) = \log\left(1 + \frac{x^2}{2\sigma^2}\right) \tag{6.47}$$

For each method, a coarse-to-fine approach is used. Starting from a coarse grid representing the image acquisitions, a first minimum of the cost function is computed. Then, the resolution is improved and a new minimum is computed using the previous one as initialization. This procedure is iterated up to the finest resolution and ensures the convexity of the cost function. Before the

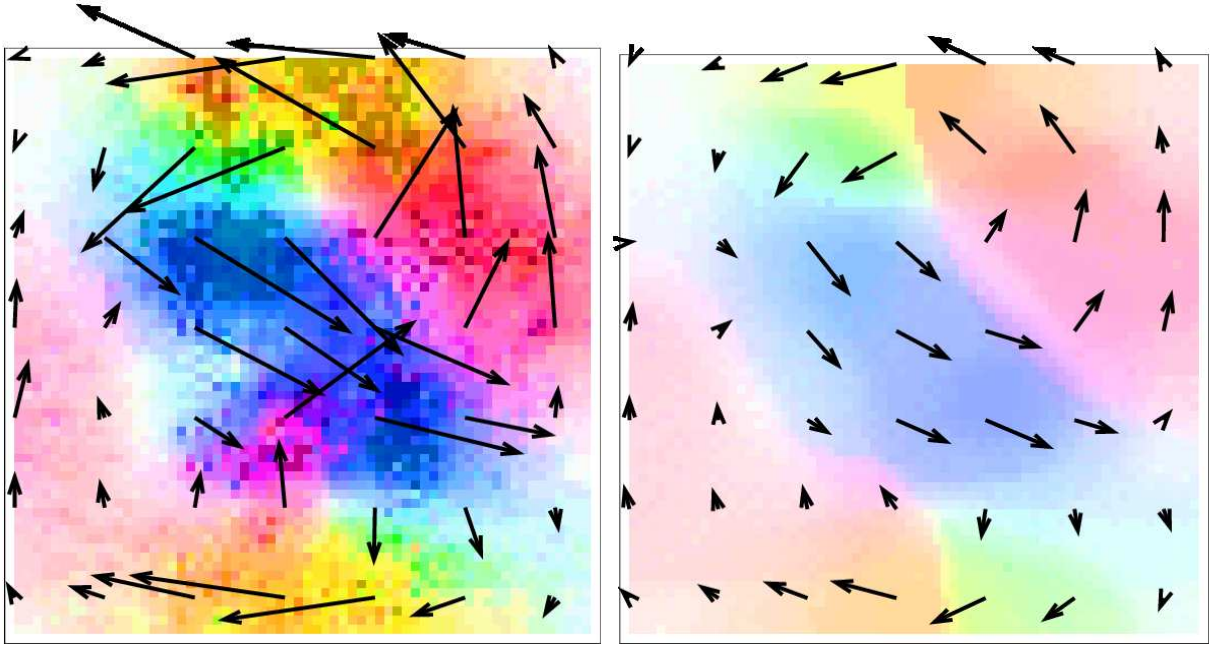

Figure 6.1 – Two members. Left: small value of λ. Right: high value of λ.

resolution is increased, the motion field at the current scale is smoothed by a median filter, whose size is also a parameter of the estimation method. It is well known that the initialization used as starting condition for an iterative minimization process highly impacts the results. Therefore, the value of the parameter smoothing the result before increasing the resolution highly impacts the quality of the solution as illustrated by Sun et al. in [11]. Another possibility to control the smoothness of the motion field obtained by one method is to include the median filtering directly in the cost function. The data term $f(\cdot)$ is then no more a local term, comparing pairwise values, but integrates, in the cost function, information over large neighborhoods. This data term has been defined by Li et al. in [72] and is weighted by an additional parameter.

Motion is then estimated between the first two images of the sequence with this whole set of methods and various parametrizations concerning the regularization weight, the norm type, the filter type (median or weighted median). An individual motion result is denoted $\mathbf{w}_0^i$ with the index i spanning the whole set of methods and parameters values. Let denote $\mathbf{w}_0 = \overline{\mathbf{w}_0^i}$ the mean of results. An uncertainty measure on the motion fields is defined by the following covariance matrix, directly computed on the estimated values:

$$B_0 = \overline{(\mathbf{w}_0^i - \mathbf{w}_0)(\mathbf{w}_0^i - \mathbf{w}_0)^T} \tag{6.48}$$

The motion field corresponding to an optimal parametrization of one of the best methods of the literature (the method of Sun et al. [11]) is then chosen for constructing the ensemble at initial date. The motion ensemble is constructed from Sun's motion field and the normal law associated to the covariance matrix B_0. One member is denoted $\mathbf{w}_0^i$ (we keep the same notation for sake of simplicity). The image component of each ensemble member is chosen equal to the first acquisition of the sequence. The corresponding background covariance error matrix is obtained by:

$$B_0^{(b)} = \overline{(\mathbf{X}_0^{(b),i} - \mathbf{X}_0^{(b)})(\mathbf{X}_0^{(b),i} - \mathbf{X}_0^{(b)})^T} \tag{6.49}$$

In order to illustrate the resulting ensemble and how members vary one from each other, an experiment is displayed on Figures 6.2 and 6.3. The first line of Figure 6.2 illustrates the first two images of the sequence and the second line gives the motion field estimated by Sun et al. between these two frames. Figure 6.3 displays members of the ensemble obtained by applying a set of optical flow methods.

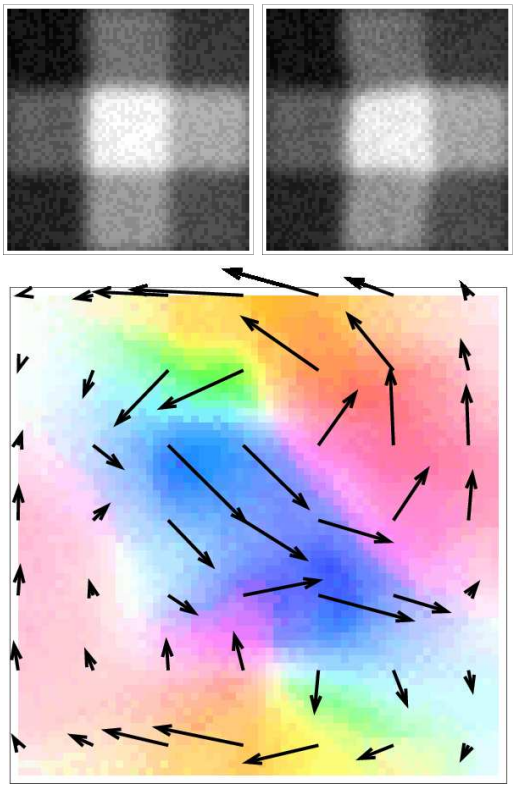

Figure 6.2 – First line: images of the sequence. Second line: mean of the motion fields ensemble.

This section discussed the design of the Image Model and the initial ensemble construction. These are the two components required for implementing the Image-based Ensemble Kalman Filter. Applying it on image data further requires the design of an observations ensemble from each image acquisition, as discussed in the next section.

6.3 Observations ensemble

Let remind that the state vector is written $\mathbf{X}_t = \begin{pmatrix} \mathbf{w}_t \\ I_t \end{pmatrix}$ with $\mathbf{w}_t$ the motion field and I_t the image function. The image function I_t satisfies the assumption that brightness values are transported by motion, as it was explained in Subsection 6.2.1. In that case, the observation operator $\mathbb{H}$ is defined as the linear projection on the second component of the state vector:

$$\mathbb{H}(\mathbf{X}_t) = I_t \tag{6.50}$$

This observation operator is used during the assimilation process in order to compare the function I_t with the real image acquisitions I_t^O: the motion field is estimated in order minimize their

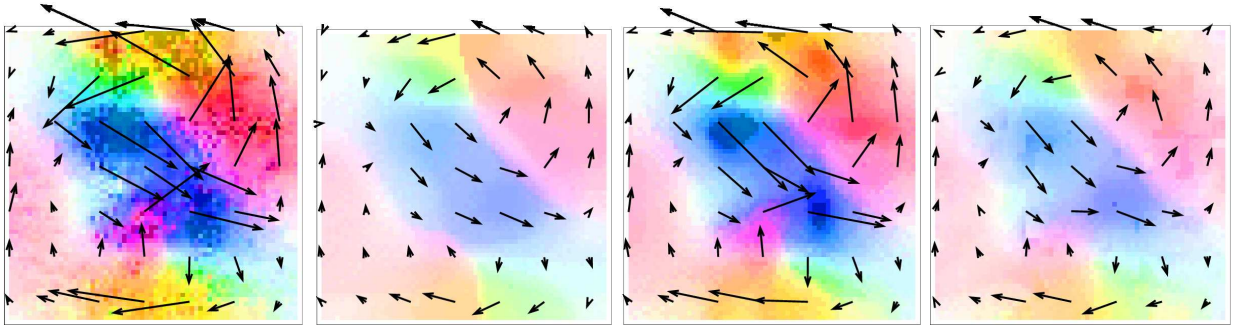

Figure 6.3 – Four members of the initial ensemble.

discrepancy.

Due to the approximation of the uncertainty matrix $B_t^{(b)}$ from a sampling by the members of the ensemble at each date t:

$$B_t^{(b)} \approx \overline{(\mathbf{X}_t^{(b),i} - \mathbf{X}_t^{(b)})(\mathbf{X}_t^{(b),i} - \mathbf{X}_t^{(b)})^T} \tag{6.51}$$

the Ensemble Kalman Filter suffers of a premature reduction in the ensemble spread. This is called shrinking in the literature. We make use of an observations ensemble in order to solve this limitation. The following demonstration, borrowed from Sakov and Oke in [61], puts in evidence that using the EnKF without any observations ensemble results in the premature shrinking of the ensemble.

We consider, at the acquisition date t, an ensemble of N_m state vectors $\mathbf{X}_t^i$, where $i = 1 \dots \mathrm{N}_m$. We also consider an ensemble of observations $\mathbf{Y}_t^i$ at the same date. Let define the perturbation matrix on the state vectors, $\mathcal{X}_\mathrm{t}$, and the perturbation matrix on the observation vectors, $\mathcal{Y}_\mathrm{t}$, at date t by:

$$\mathcal{X}_\mathrm{t} = \left[\left(\mathbf{X}_t^1 - \overline{\mathbf{X}_t^i}\right) \dots \left(\mathbf{X}_t^i - \overline{\mathbf{X}_t^i}\right) \dots \left(\mathbf{X}_t^{\mathrm{N}_m} - \overline{\mathbf{X}_t^i}\right)\right] \tag{6.52}$$

$$\mathcal{Y}_\mathrm{t} = \left[\left(\mathbf{Y}_t^1 - \overline{\mathbf{Y}_t^i}\right) \dots \left(\mathbf{Y}_t^i - \overline{\mathbf{Y}_t^i}\right) \dots \left(\mathbf{Y}_t^{\mathrm{N}_m} - \overline{\mathbf{Y}_t^i}\right)\right] \tag{6.53}$$

with $\overline{\mathbf{X}_t^i}$ and $\overline{\mathbf{Y}_t^i}$ respectively defined by Equation (6.1) and Equation (6.2). Using these notations, it becomes possible to rewrite the analysis equation, Equation (6.36), as an equation on the perturbations:

$$\mathcal{X}_\mathrm{t}^{(\mathrm{a}),\mathrm{i}} = \mathcal{X}_\mathrm{t}^{(\mathrm{b}),\mathrm{i}} + \mathrm{K}(\mathcal{Y}_\mathrm{t} - \mathbb{H}\mathcal{X}_\mathrm{t}^{(\mathrm{b})}) \tag{6.54}$$

If using Equation (6.39) in order to compute the error covariance carried by the ensemble after the analysis, it comes:

$$B_t^{(a)} = \frac{1}{\mathrm{N}_m - 1} \mathcal{X}_\mathrm{t}^{(\mathrm{a})} \mathcal{X}_\mathrm{t}^{(\mathrm{a})^\mathrm{T}} \tag{6.55}$$

Replacing the perturbation matrix $\mathcal{X}_\mathrm{t}^{(\mathrm{a})}$ in Equation (6.55) with its expression given in Equation (6.54) leads to the following formulation:

$$B_t^{(a)} = \frac{1}{\mathrm{N}_m - 1} \left[\mathcal{X}_\mathrm{t}^{(\mathrm{b})} + \mathrm{K}(\mathcal{Y}_\mathrm{t} - \mathbb{H}\mathcal{X}_\mathrm{t}^{(\mathrm{b})})\right] \left[\mathcal{X}_\mathrm{t}^{(\mathrm{b})} + \mathrm{K}(\mathcal{Y}_\mathrm{t} - \mathbb{H}\mathcal{X}_\mathrm{t}^{(\mathrm{b})})\right]^T \tag{6.56}$$

After expansion of Equation (6.56), it comes:

$$B_t^{(a)} = B_t^{(b)} - B_t^{(b)}\mathbb{H}^T K^T - K\mathbb{H}B_t^{(b)} + K\mathbb{H}B_t^{(b)}\mathbb{H}^T K^T + \frac{1}{\mathrm{N}_m - 1}\left[K\mathcal{Y}_\mathrm{t}\mathcal{Y}_\mathrm{t}^\mathrm{T}\mathrm{K}^\mathrm{T} + (Id - \mathrm{K}\mathbb{H})\mathcal{X}_\mathrm{t}^{(\mathrm{b})}\mathcal{Y}_\mathrm{t}^\mathrm{T}\mathrm{K}^\mathrm{T} + \mathrm{K}\mathcal{Y}_\mathrm{t}\mathcal{X}_\mathrm{t}^{(\mathrm{b})^\mathrm{T}}(Id - \mathbb{H}^\mathrm{T}\mathrm{K}^\mathrm{T})\right] \quad (6.57)$$

where Id is the identity matrix.

If the observation is unique (if we are not considering having an observation ensemble), this nullifies $\mathcal{Y}_\mathrm{t}$ and therefore nullifies the second line of Equation (6.57). The background error covariance matrix $B_t^{(a)}$ is then rewritten, after factorization, as:

$$B_t^{(a)} = (Id - K\mathbb{H})B_t^{(b)} - (Id - K\mathbb{H})B_t^{(b)}\mathbb{H}^T K^T \quad (6.58)$$

which can be compared to:

$$B_t^{(a)} = (Id - K\mathbb{H})B_t^{(b)} \quad (6.59)$$

which is the value of the covariance matrix given by the Kalman filter after the analysis step.

It becomes clear that, if not using an observations ensemble, the error covariance matrix of the analysis, computed on the members, suffers from the subtraction of $(Id - K\mathbb{H})B_t^{(b)}\mathbb{H}^T K^T$ compared to the analytical one, given by Equation (6.59).

Let assume having an ensemble of observations $\mathbf{Y}_t^i$ at date t, correctly sampling, in a statistical sense, the observation error covariance matrix R_t. It comes, by definition of $\mathcal{Y}_\mathrm{t}$:

$$\frac{1}{\mathrm{N}_m - 1}\mathcal{Y}_\mathrm{t}\mathcal{Y}_\mathrm{t}^\mathrm{T} = \mathrm{R}_\mathrm{t} \quad (6.60)$$

Let also assume R_t independent from the background covariance matrix $B_t^{(b)}$. This assumption leads, considering the perturbation matrix on the state vector $\mathcal{X}_\mathrm{t}^{(\mathrm{b})}$ and the perturbation matrix on the observation vector $\mathcal{Y}_\mathrm{t}^{(\mathrm{b})}$, to the equation:

$$\frac{1}{\mathrm{N}_m - 1}\mathcal{X}_\mathrm{t}^{(\mathrm{b})}\mathcal{Y}_\mathrm{t}^\mathrm{T} = 0 \quad (6.61)$$

where 0 is a null matrix. Using the two assumptions expressed by Equations (6.60, 6.61) in Equation (6.57) leads to:

$$B_t^{(a)} = B_t^{(b)} - B_t^{(b)}\mathbb{H}^T K^T - K\mathbb{H}B_t^{(b)} + K\mathbb{H}B_t^{(b)}\mathbb{H}^T K^T + KR_tK^T \quad (6.62)$$

$$= B_t^{(b)} - K\mathbb{H}B_t^{(b)} - B_t^{(b)}\mathbb{H}^T K^T + K\left(\mathbb{H}B_t^{(b)}\mathbb{H}^T + R_t\right)K^T \quad (6.63)$$

By using the definition of the Kalman gain, Equation (6.37), in Equation (6.63) it comes:

$$B_t^{(a)} = B_t^{(b)} - K\mathbb{H}B_t^{(b)} - B_t^{(b)}\mathbb{H}^T K^T + B_t^{(b)}\mathbb{H}^T K^T \quad (6.64)$$

$$= B_t^{(b)} - K\mathbb{H}B_t^{(b)} \quad (6.65)$$

$$= (Id - K\mathbb{H})B_t^{(b)} \quad (6.66)$$

Equation (6.66), defining the value of $B_t^{(a)}$ in the ensemble Kalman filter if an ensemble of observations is used, leads to an approximation of the analytical value of $B_t^{(a)}$, Equation (6.59), in a statistical sense.

As previously justified, observations ensembles are mandatory in the experiments in order to avoid the shrinking effect that occurs when computing the analysis. In the case of motion estimation from images, the formulation of the observations ensemble associated to one image acquisition is more or less straightforward. Metadata that are available on the sensor characteristics allow the user to model the acquisition noise as an additive zero-mean Gaussian noise, whose standard deviation value is usually included in the information file associated to the sensor. This is written as:

$$I_t^O = I_t^{(r)} + \mathcal{N}(0, R_t) \tag{6.67}$$

where I_t^O is the acquired image of the unknown truth $I_t^{(r)}$. The image ensemble associated to I_t^O is then obtained with the same heuristics. Each member I_t^i is computed by adding a Gaussian noise to the acquired image I_t^O:

$$I_t^i = I_t^O + \mathcal{N}(0, R_t) \tag{6.68}$$

For an optimal implementation of the ensemble Kalman filter, as the one described by Evensen in [68], which relies on a singular value decomposition of the matrix $\mathbb{H}\mathcal{X}_t^{(b)} + \mathcal{Y}_t$, it is necessary to use as many members in the observations ensemble and in the state vectors ensemble.

6.4 Explicit localization

As seen in Section 6.3, an observations ensemble is required to correctly approximate the uncertainty covariance matrix after computing the analysis. It is, however, not the only point where the EnKF suffers from sampling problems due to the approximation of the background uncertainty matrix $B_t^{(b)}$ by:

$$B_t^{(b)} \approx \overline{(\mathbf{X}_t^{(b),i} - \mathbf{X}_t^{(b)})(\mathbf{X}_t^{(b),i} - \mathbf{X}_t^{(b)})^T} \tag{6.69}$$

Equation (6.69) usually includes spurious covariances. Therefore, a couple of pixels that should have been uncorrelated will get a high covariance value due to the approximation of $B_t^{(b)}$ with Equation (6.69). These spurious values can appear anywhere in the background error covariance matrix. Moreover, considerations on which values are spurious or not are highly dependent from the application and from the users idea of the analytical matrix $B_t^{(b)}$.

An explicit localization method is first used in order to limit the impact of these spurious covariances. The method is based on a localization function ρ that corrects the spurious covariances by multiplying the covariance values with the values from the localization function. Often, these spurious covariances link pixels that should be independent, because being too far one from each other or belonging to different structures of the image. The background error covariance matrix is then supposed to recover some of its analytical properties by this multiplication. Let ρ denote the localization matrix corresponding to the localization function ρ, and $\circ$ the point-wise matrix product, also known as the Hadamard product. Defining:

$$L_t^{(b)} = \rho \circ B_t^{(b)} \tag{6.70}$$

and replacing $B_t^{(b)}$ by $L_t^{(b)}$ in the analysis equations, Equations (6.36) and (6.37), it comes:

$$\mathbf{X}_t^{(a),i} = \mathbf{X}_t^{(b),i} + K_L \left(\mathbf{Y}(k) - \mathbb{H}\mathbf{X}_t^{(b),i} \right) \tag{6.71}$$

that computes the analysis $\mathbf{X}_t^{(a),i}$ of member i at date t, with:

$$K_L = L_t^{(b)} \mathbb{H}^T \left(\mathbb{H} L_t^{(b)} \mathbb{H}^T + R_t \right)^{-1} \tag{6.72}$$

The localization process is applied before computing the analysis from the background values and the observations.

There are two main groups of methods for defining the localization matrix ρ. In the first one, the function ρ, denoted ρ_d (where the subscript d stands for *d*istance), is defined according to the distance between pixels, as in [64]. ρ_d is thus designed so that pixels separated by a distance higher than a given threshold get an almost null correlation. Let $\mathbf{p}_1$ and $\mathbf{p}_2$ denote two pixels of the image domain, ρ_d depends on their distance $d_{12} = ||\mathbf{p}_1 - \mathbf{p}_2||_2$ and is defined by:

$$\rho_d(\mathbf{p}_1, \mathbf{p}_2) = \left(1 + \frac{d_{12}}{a_d}\right) \exp\left(-\frac{d_{12}}{a_d}\right) \tag{6.73}$$

The parameter a_d, involved in Equation (6.73), is called the decorrelation distance. The values of ρ_d, depending on d_{12}, are displayed on Figure 6.5 for three different values of a_d.

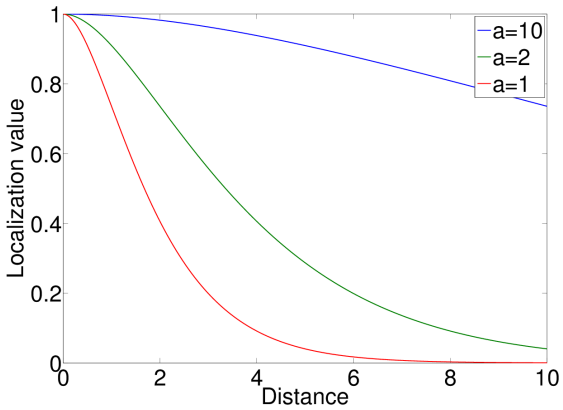

Figure 6.4 – Values of ρ_d, for $a_d = 1$, 2, 10, as a function of the distance between pixels.

In the second group of methods, the matrix ρ also nullifies the correlation between pixels that belong to different structures. The matrix ρ_s (where the subscript s stands for *s*imilarity) depends on pixels similarities. Let denote I_1 and I_2 the gray values of pixels $\mathbf{p}_1$ and $\mathbf{p}_2$ and $s_{12} = \sqrt{(I_1 - I_2)^2}$. The function ρ_s is defined by:

$$\rho_s(\mathbf{p}_1, \mathbf{p}_2) = \left(1 + \frac{s_{12}}{a_s}\right) \exp\left(-\frac{s_{12}}{a_s}\right) \tag{6.74}$$

where a_s is the decorrelation parameter according to the similarity between pixels.

The localization matrix ρ, used in Equation (6.70), is then defined as the point-wise product of matrices ρ_d and ρ_s. Consequently, the correlation values between pixels that are far apart or belong to different structures are almost null.

In order to illustrate the localization process, Figure 6.5 displays, from left to right, the image and the reference point (*in red)*, the binary representation of the structure, the localization value between the reference pixel and every other pixel of the domain considering only the distance, the localization value between the reference pixel and every other pixel of the domain considering the distance and the similarity between pixels. It can be seen that the values of the localization function using the similarity function have lower values for pixels which are not belonging to the structure.

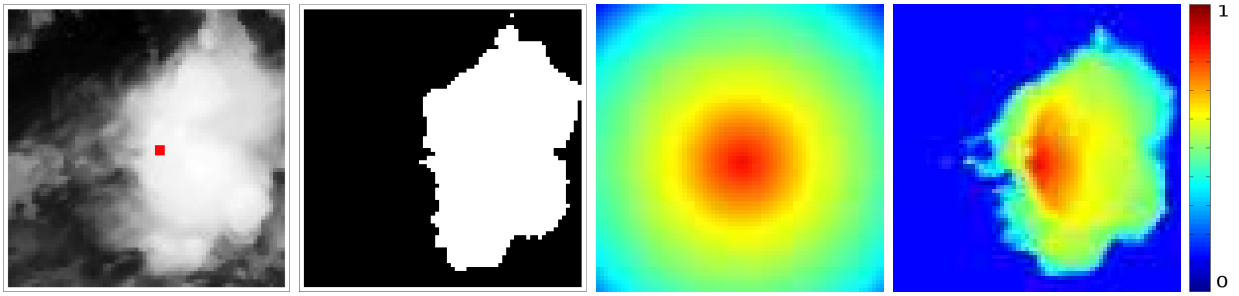

Figure 6.5 – From left to right: image; binary representation of the structure; localization function between the red point and each other pixel based on distance; localization function between the red point and each other pixel based on distance and similarity.

6.5 Implementation and localization

This section will only focus on the cost of the matrix inversion required in the computation of the Kalman gain:

$$K = B_t^{(b)} \mathbb{H}^T \left(\mathbb{H} B_t^{(b)} \mathbb{H}^T + R_t \right)^{-1} \tag{6.75}$$

In fact, this inversion is the bottleneck of the EnKF approach while using localization. Let first consider the implementation without localization as an efficient localization method should preserve, as possible, the same complexity than the original algorithm.
The ensemble Kalman filter can be efficiently implemented by applying a **S**ingular **V**alue **D**ecomposition (**SVD**) for computing the inversion needed in the Kalman gain of Equation (6.75). Considering having an ensemble of observations $\mathbf{Y}_t^i$ correctly sampling the covariance matrix R_t, Equation (6.60), and assuming the independence between the members $\mathbf{Y}_t^i$ and $\mathbf{X}_t^j$, Equation (6.61), it comes, by definition of the state vector perturbation matrix $\mathcal{X}_\mathrm{t}$ and of the observation vector perturbation matrix $\mathcal{Y}_\mathrm{t}$, that:

$$\mathbb{H} B_t^{(b)} \mathbb{H} + R_t = \left(\mathbb{H}\mathcal{X}_\mathrm{t} + \mathcal{Y}_\mathrm{t} \right) \left(\mathbb{H}\mathcal{X}_\mathrm{t} + \mathcal{Y}_\mathrm{t} \right)^T \tag{6.76}$$

Therefore, for inverting $\left(\mathbb{H} B_t^{(b)} \mathbb{H} + R_t \right)$, it is only necessary to compute the SVD of $(\mathbb{H}\mathcal{X}_\mathrm{t} + \mathcal{Y}_\mathrm{t})$ rather than computing it on $\left(\mathbb{H} B_t^{(b)} \mathbb{H} + R_t \right)$. After computation of the singular values and associated matrices, it comes by definition:

$$\mathbb{H}\mathcal{X}_\mathrm{t} + \mathcal{Y}_\mathrm{t} = \mathrm{U}\Sigma\mathrm{V}^\mathrm{T} \tag{6.77}$$

where U and V are orthogonal matrices and Σ is a diagonal matrix. Let remark that $\mathbb{H}\mathcal{X}_\mathrm{t}$ and $\mathcal{Y}_\mathrm{t}$ are rectangular matrices. The minimum of their number of columns and number of rows defines the rank of $(\mathbb{H}\mathcal{X}_\mathrm{t} + \mathcal{Y}_\mathrm{t})$. This corresponds to the number of non null values in Σ. In our application, the number of members N_m being much smaller than the number of variables in the observation vector $\mathrm{N}_\mathbf{Y}$, Σ has only N_m non null values. The inversion is then efficiently computed by considering that:

$$\left(\mathbb{H} B_t^{(b)} \mathbb{H} + R_t \right)^{-1} = U \Sigma^{-2} U^T \tag{6.78}$$

The computational cost associated to this inversion mainly comes from the SVD and is proportional to $\mathrm{N}_\mathbf{Y} {\mathrm{N}_m}^2$.

Let now consider having a localization function ρ and its associated matrix ρ of size $N_{\mathbf{Y}}N_{\mathbf{Y}}$. Let regard the formulation of ρ for two pixels $\mathbf{p}_1$ and $\mathbf{p}_2$:

$$\rho(\mathbf{p}_1, \mathbf{p}_2) = \rho_d(\mathbf{p}_1, \mathbf{p}_2)\rho_s(\mathbf{p}_1, \mathbf{p}_2) \tag{6.79}$$

where ρ_d is defined by Equation (6.73) and ρ_s by Equation (6.74). If the two lines or rows, corresponding to $\mathbf{p}_1$ and $\mathbf{p}_2$, are linearly dependent in the matrix ρ, it comes:

$$\rho(\mathbf{p}_1, \mathbf{p}) \propto \rho(\mathbf{p}_2, \mathbf{p}) \quad \forall \mathbf{p} \in \Omega \tag{6.80}$$

which is impossible to obtain considering the distance and the similarity between pixels on real image acquisitions. Therefore, ρ has the rank value $N_{\mathbf{Y}}$.

It is then impossible to decompose $\left(\mathbb{H}(\rho \circ B_t^{(b)})\mathbb{H} + R_t\right)$ into the product of two lower rank matrices as done in Equation (6.76). The inversion required by the Kalman gain, Equation (6.75), is then applied on the whole rank matrix $\left(\mathbb{H}(\rho \circ B_t^{(b)})\mathbb{H} + R_t\right)$, leading to two drawbacks. First, the matrix to be inverted is of size ${N_{\mathbf{Y}}}^2$, which is unaffordable to store in memory (hundred of *Go* for a 500×500 pixels image). Second, inverting such a matrix requires ${N_{\mathbf{Y}}}^3$ operations, leading to a prohibitive computational and storage costs, even considering iterative approaches implemented in parallel, as proposed by the library PLASMA [73]. A solution to this issue is introduced in the following conclusion and fully described in Chapter 7.

6.6 Conclusion

This chapter described a motion estimation method based on the Ensemble Kalman Filter. We first demonstrated how to derive the Kalman equations starting from the Bayes' formula. We then showed that it is necessary, for large state vectors, to use the ensemble Kalman approximation. A few implementation details, as the initial ensemble design and some reminders about the data assimilation components were given. Last, two major drawbacks of the EnKF were put in evidence, before to discuss solutions for effective implementation:

- the shrinking of the ensemble when computing the analysis. It is avoided thanks to an ensemble of observations.
- the spurious covariances that are due to sampling errors. Their impact is limited by localization functions.

Localization is necessary for an accurate estimation and for the uncertainty matrix after analysis to be valid. However, as Explained in Section 6.5, the proposed localization method will not be used in an operational context, as it strongly increases the computational cost of the algorithm. The increase is proportional to the cube of the size of the observation vector $\mathbf{Y}$. In fact, the matrix inversion to be found in the Kalman gain, being affordable if Evensen's implementation [68] is followed, becomes prohibitive in the context of explicit localization. This drawback, due to the localization function, leads to the necessity to define a more suitable localization method. A domain decomposition approach has then been designed and is fully described in Chapter 7.

Chapter 7

Domain decomposition

The ensemble Kalman filter, as seen in Chapter 6, is based on a sampling of the background covariance matrix $B_t^{(b)}$ thanks to the ensemble members $\mathbf{X}_t^i$:

$$B_t^{(b)} \approx \overline{(\mathbf{X}_t^{(b),i} - \mathbf{X}^{(b)})(\mathbf{X}_t^{(b),i} - \mathbf{X}_t^{(b)})^T} \tag{7.1}$$

where $\mathbf{X}^{(b)}$ is the mean of the ensemble state vectors $\mathbf{X}_t^{(b),i}$. This sampling allows to get rid of the covariance propagation problem, which is the main drawback of the original Kalman filter. However, sampling the covariance between pixels of the image domain by a limited number of members creates spurious covariances between locations, which should otherwise be almost uncorrelated. This is the case, for instance, for pixels which are neighbors, but belonging to different structures, or for those whose distance is large enough. As described in Chapter 6, the explicit localization techniques limit the impact of these spurious covariances and almost nullify them. This localization, as described by Mitchell et al. in [74], is applied by an elementwise matrix product between the covariance matrix $B_t^{(b)}$ and a localization matrix ρ, containing null values for these pairs of pixels. The localization matrix ρ is defined from a localization function ρ expressing the user assumptions. However, this technique increases the rank of the covariance matrix $B_t^{(b)}$ and therefore drastically increases the computational cost of the ensemble Kalman filter. If the size of the image domain Ω is large, with more than ten thousand pixels (which corresponds to a small 100×100 image), this comes to the point where the method described in Section 6.4 of Chapter 6 becomes unaffordable. This solution is then out of scope for estimating motion from an image sequence in an operational context. Another approach for defining the localization method has been described by Hunt et al. [75] and Nerger et al. [76]. It consists in decomposing the studied domain Ω in subdomains and independently computing an analysis on each of them.

This chapter focuses on this alternative for localization in the studied context of motion estimation from an image sequence. Section 7.1 describes the principles of localization with domain decomposition and its impact on the Ensemble Kalman Filter. A reminder of the notations used in Chapter 6 and some additional ones are given in Subsection 7.1.1. In order to introduce the principles of localization with domain decomposition, a simple case of such approach is fully described in Subsection 7.1.2. This allows to initiate the discussion and introduce the main ideas, which will be further developed in Subsection 7.1.3 with a more realistic approach of localization with domain decomposition.

Then, Section 7.2 describes a method for decomposing the domain in subdomains, accordingly with the structures that are displayed on the image acquisitions. This allows that the only covari-

ances that are non null, in the background error covariance matrix $B_t^{(b)}$, are those between pixels, which are close or similar.

7.1 Localization and domain decomposition

The explicit localization method may be compared to the localization technique with domain decomposition, when they are applied to an Ensemble Kalman Filter. For sake of clarity, let first recall some notations and introduce a few new ones. Then, let focus on a simple case and discuss the major principles of the domain decomposition approach.

7.1.1 Notations and reminders

Let begin the description of the localization with domain decomposition by a few reminders and the introduction of some notations, which are mandatory for a clear discussion on the subject.

1. The value of the state vector $\mathbf{X}_t$, at date t, is denoted $\mathbf{X}_t = \begin{pmatrix} \mathbf{w}_t \\ I_t \end{pmatrix}$, where $\mathbf{w}_t = \begin{pmatrix} u_t \\ v_t \end{pmatrix}$ is the motion field and I_t is an image function. Each function u_t, v_t and I_t, at date t, is defined on the image domain Ω. The entire chapter focuses only on the manner of computing a single analysis, at a given date t. For simplifying the notations the subscript t is then suppressed and every variable is considered at the same date t.

2. As previously, the goal is to obtain an estimation of the state vector $\mathbf{X}^{(a)}$ by combining a prior knowledge $\mathbf{X}^{(b)}$ (resulting from an integration of the analysis obtained at the previous time step) with an observation vector $\mathbf{Y}$.

3. The background error is supposed Gaussian and zero-mean, and its covariance matrix is denoted $B^{(b)}$. In the implementation of EnKF, $B^{(b)}$ is computed from the ensemble members $\mathbf{X}^{(b),i}$ by:
$$B^{(b)} \approx \overline{(\mathbf{X}^{(b),i} - \mathbf{X}^{(b)})(\mathbf{X}^{(b),i} - \mathbf{X}^{(b)})^T} \tag{7.2}$$
where $\mathbf{X}^{(b)} = \overline{\mathbf{X}^{(b),i}}$ denotes the ensemble mean.

4. Let recall that the observation vector $\mathbf{Y}$ is composed of the values of the acquisition I^O. The observation error has to be specified. It is supposed Gaussian and zero-mean, and its covariance matrix, called the observation error covariance matrix, is denoted R. R has always been considered as a diagonal matrix for all applications described in this document, whether it was for the variational methods or the sequential one. This assumption is kept in this chapter.

5. Let remind that the observation operator $\mathbb{H}$ is a projection on the third component of the state vector:
$$\mathbb{H}\mathbf{X} = I \tag{7.3}$$

6. The discrete image domain Ω will be decomposed in subdomains, denoted D_i for $i = 1, \ldots, \mathrm{K}_D$, where K_D is the total number of subdomains. The assumption is that the decomposition into

D_i spans the whole image domain Ω without overlapping, which is mathematically written as:

$$\bigcup_{i=1}^{\mathrm{K}_D} D_i = \Omega \tag{7.4}$$

and

$$D_i \cap D_j = 0 \quad \forall (i,j) \tag{7.5}$$

7. Restriction operators are needed to describe the localization with domain decomposition. Let introduce the notation $\mathcal{R}_{D_i}$ that restricts any function defined on the domain Ω on one function defined on the subdomain D_i. Let also introduce the restriction operator $\mathcal{R}_D$, which transforms any function defined on the whole acquisition domain as a vector of the restrictions onto all subdomains:

$$\mathcal{R}_D = \begin{bmatrix} \mathcal{R}_{D_1} \\ \vdots \\ \mathcal{R}_{D_i} \end{bmatrix} = \begin{bmatrix} Id_{D_1} & 0 & \dots & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & \dots & 0 & Id_{D_{\mathrm{K}_D}} \end{bmatrix} \tag{7.6}$$

where Id_{D_i} is the identity matrix defined on the subdomain D_i. It comes, from Equation (7.6), that applying the restriction operator $\mathcal{R}_D$ on any function defined on the image domain Ω, for instance u, leads to the restricted vector u_D defined by:

$$\mathcal{R}_D u = \begin{bmatrix} \mathcal{R}_{D_1} u \\ \vdots \\ \mathcal{R}_{D_i} u \end{bmatrix} = \begin{bmatrix} u_{D_1} \\ \vdots \\ u_{D_i} \end{bmatrix} \tag{7.7}$$

8. When localizing the analysis equation with domain decomposition, the local equivalent of $B^{(b)}$, for the subdomain decomposition D, will be denoted $B_D^{(b)}$ and defined by:

$$B_D^{(b)} = \mathcal{R}_D B^{(b)} \mathcal{R}_D^T \tag{7.8}$$

Similarly, the restriction of R, denoted R_D is defined by:

$$R_D = \mathcal{R}_D R \mathcal{R}_D^T \tag{7.9}$$

9. In Subsection 7.1.3, two different domain decompositions are needed.

   A. The first one concerns the analysis. This decomposition is denoted D. Its subdomains are denoted D_i and called analysis subdomains. The number of pixels in the subdomain D_i is denoted N_{D_i}. Let remark that, according to Equations (7.4) and (7.5), the union of every D_i spans the whole image domain Ω without overlapping, it comes:

$$\mathrm{N}_\Omega = \mathrm{N}_D = \sum_{i=1}^{\mathrm{K}_D} \mathrm{N}_{D_i} \tag{7.10}$$

   where N_Ω is the number of pixels in Ω.

B. The second decomposition concerns the observations and is denoted E. Its subdomains are called the observation subdomains and denoted E_i. The number of pixels in E_i is denoted N_{Ei}. The observation subdomain E_i is associated to the analysis subdomain D_i, therefore the number of observation subdomains is equal to the number N_D of analysis subdomains. It will be shown that, for smoothness of the estimation, the E_i should overlap, such that the analysis computed on D_i uses observations from its neighboring analysis subdomains. To do so, the analysis subdomain D_i is first computed and enlarged, by a certain number of pixels in every direction, in order to obtain E_i. Let denote N_E the number of pixels defined by:

$$\mathrm{N}_E = \sum_{i=1}^{\mathrm{K}_D} \mathrm{N}_{Ei} \tag{7.11}$$

As the subdomains E_i are overlapping, the number of pixels N_E, corresponding to the decomposition in observation subdomains E_i, is higher than the number of pixels N_D in the decomposition D. Let define the number of overlapping pixels N_{Ov}, by:

$$\mathrm{N}_{Ov} = \mathrm{N}_E - \mathrm{N}_D \tag{7.12}$$

10. Having defined the two different domain decompositions, D and E, involved in the localization process, let denote their respective restriction by the operators $\mathcal{R}_D$ and $\mathcal{R}_E$.

Let recall that the values of one function, defined on the discrete image domain Ω, are described by a vector, on a line by line, column by column coding. However, these values are displayed as images for a better understanding. The transition from one representation to the other, being trivial, is no more notified in the following. For the precise description of the transition, the Reader should refer to Section 1.1 of Chapter 1

The notations required for the description of the localization with domain decomposition have been defined. The next subsection will describe, on a basic case, the idea behind this approach of localization. The objective is to clarify the links between the explicit localization and the localization with domain decomposition.

7.1.2 Introduction to domain decomposition

Let introduce the localization with domain decomposition by considering its simplest formulation. We consider that the studied domain Ω is decomposed in four subdomains D_1, D_2, D_3 and D_4, displayed on Figure 7.1. Such decomposition allows to introduce and discuss in a simple way the main principles of the localization with domain decomposition.

Let express the links between the approach of localization based on the domain decomposition and the explicit localization method, which has been discussed in Section 6.4 of Chapter 6, that is based on the localization function ρ. Let first rewrite the background error covariance matrix $B^{(b)}$

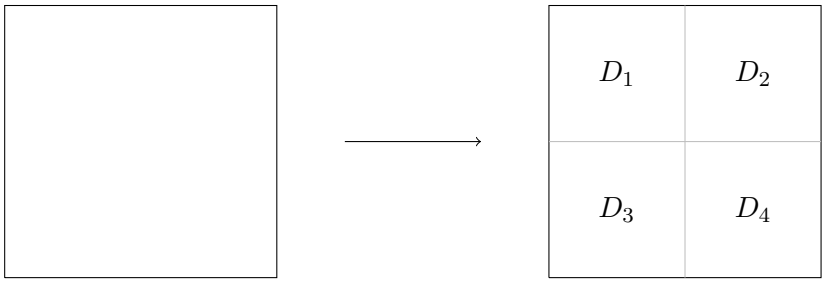

Figure 7.1 – Decomposition of Ω into four subdomains D_i.

by focusing on the covariances between the subdomains D_i:

$$B^{(b)} = \begin{bmatrix} B^{(b)}_{D_{1,1}} & B^{(b)}_{D_{1,2}} & B^{(b)}_{D_{1,3}} & B^{(b)}_{D_{1,4}} \\ B^{(b)}_{D_{1,2}} & B^{(b)}_{D_{2,2}} & B^{(b)}_{D_{2,3}} & B^{(b)}_{D_{2,4}} \\ B^{(b)}_{D_{1,3}} & B^{(b)}_{D_{2,3}} & B^{(b)}_{D_{3,3}} & B^{(b)}_{D_{3,4}} \\ B^{(b)}_{D_{1,4}} & B^{(b)}_{D_{2,4}} & B^{(b)}_{D_{3,4}} & B^{(b)}_{D_{4,4}} \end{bmatrix} \tag{7.13}$$

where $B^{(b)}_{D_{i,j}}$ represents the covariances between the pixels of the subdomain D_i and the pixels of the subdomain D_j.
Rewriting the background error covariance matrix defined by Equation (7.13) into its local form $B^{(b)}_D$ defined by Equation(7.8), it comes:

$$B^{(b)}_D = \mathcal{R}_D B^{(b)} \mathcal{R}_D^T \begin{bmatrix} B^{(b)}_{D_{1,1}} & 0 & 0 & 0 \\ 0 & B^{(b)}_{D_{2,2}} & 0 & 0 \\ 0 & 0 & B^{(b)}_{D_{3,3}} & 0 \\ 0 & 0 & 0 & B^{(b)}_{D_{4,4}} \end{bmatrix} \tag{7.14}$$

To draw a parallel between the explicit localization and the localization by domain decomposition, let define a new localization function ρ, as:

$$\rho(\mathbf{p}_1, \mathbf{p}_2) = \begin{cases} 1 & \text{if } \mathbf{p}_1 \in D_i \text{ and } \mathbf{p}_2 \in D_i \\ 0 & \text{otherwise} \end{cases} \tag{7.15}$$

and let denote ρ the associated localization matrix. The equation of the Ensemble Kalman Filter, associated with the explicit localization function, as explained in Section 6.4, is defined by:

$$\mathbf{X}^{(a),i} = \mathbf{X}^{(b),i} + K_L(\mathbf{Y}(k) - \mathbb{H}\mathbf{X}^{(b),i}) \tag{7.16}$$

where the gain K_L is:

$$K_L = L^{(b)}\mathbb{H}^T\left(\mathbb{H}L^{(b)}\mathbb{H}^T + R\right)^{-1} \tag{7.17}$$

and $L^{(b)}$, by denoting $\circ$ the elementwise matrix product, is defined by:

$$L^{(b)} = \rho \circ B^{(b)} \tag{7.18}$$

On another hand, let write the equation of the Ensemble Kalman Filter, associated to a localization with domain decomposition. By using Equations (7.8) and (7.9), the analysis equation is given by:

$$\mathbf{X}^{(a),i} = \mathbf{X}^{(b),i} + K_D(\mathcal{R}_D\mathbf{Y}(k) - \mathbb{H}\mathcal{R}_D\mathbf{X}^{(b),i}) \tag{7.19}$$

where:

$$K_D = \mathcal{R}_D B^{(b)} \mathcal{R}_D^T \mathbb{H}^T \left(\mathbb{H}\mathcal{R}_D B^{(b)} \mathcal{R}_D^T \mathbb{H}^T + \mathcal{R}_D R \mathcal{R}_D^T\right)^{-1} \tag{7.20}$$

Considering that the domain decomposition D spans the entire image domain Ω, that the D_i are not overlapping and that R is diagonal, Equation (7.20) can be rewritten as:

$$K_D = B_D^{(b)} \mathbb{H}^T \left(\mathbb{H} B_D^{(b)} \mathbb{H}^T + R\right)^{-1} \tag{7.21}$$

If we consider the domain decomposition of Figure 7.1 and the function ρ defined by Equation (7.15), the two background error covariance matrix obtained after localization, $B_D^{(b)}$ and $L^{(b)}$ are equal. Therefore, the two Kalman gains K_D from Equation (7.21) and K_L from Equation (7.17) are equals and the two analysis values, defined by Equations (7.16) and (7.19), are the same.

The approach of localization with domain decomposition shows some advantages over the explicit localization when applied to the Ensemble Kalman Filter. The Reader should notice that $B_D^{(b)}$ is a block diagonal matrix and R is a diagonal matrix. Therefore the matrix to be inverted in the Kalman gain, defined by Equation (7.21), is block diagonal. The inverse being equal to the matrix composed of the inverse of each block, it can be computed by independent processors. Consequently, the computational time can be reduced without any difficulty.

As stated in Subsection 7.1.1, the number of pixels N_Ω of the discrete image domain Ω is equal to the number of pixels N_D considered by the decomposition D. Let recall that the domain is decomposed in K_D subdomains D_i, with N_{Di} pixels each, leading to:

$$\mathrm{N}_D = \sum_{i=1}^{\mathrm{K}_D} \mathrm{N}_{Di} \tag{7.22}$$

The computational cost of the matrix inversion, required when computing the Kalman gain in Equation (7.17), is equal to $O({\mathrm{N}_D}^3)$, if an explicit localization method is applied, as described in Section 6.4 of Chapter 6. If the ensemble Kalman filter relies on a domain decomposition, the number of computations for inverting the matrix involved in the Kalman gain, Equation (7.21), is equal to $O(\sum_{i=1}^{\mathrm{K}_D} {\mathrm{N}_{Di}}^3)$, which is less than the cost of the explicit localization $O({\mathrm{N}_D}^3)$, as:

$${\mathrm{N}_D}^3 = (\sum_{i=1}^{\mathrm{K}_D} \mathrm{N}_{Di})^3 \gg \sum_{i=1}^{\mathrm{K}_D} {\mathrm{N}_{Di}}^3 \tag{7.23}$$

However, the localization with domain decomposition, as it has been described until now, presents the major drawback that two pixels, located on both sides of the boundary between neighboring subdomains, are totally uncorrelated. As a result, the analysis is discontinuous, as illustrated by Figure 7.2. With this rough domain decomposition, each subdomain gets independently its own

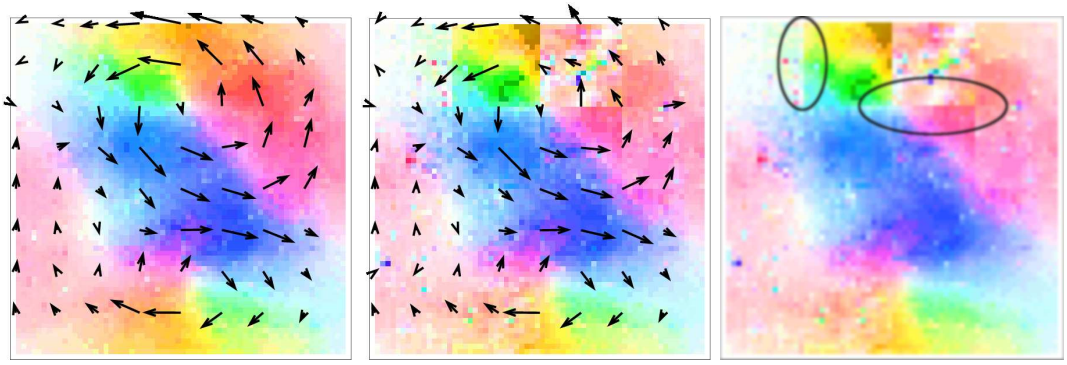

Figure 7.2 – Results with non overlapping subdomains, from left to right: Motion background $\mathbf{w}^{(b)}$; Motion analysis $\mathbf{w}^{(a)}$; $\mathbf{w}^{(a)}$ with highlights on discontinuities between subdomains.

analysis. No process is applied to ensure the continuity at the boundaries between these subdomains. This is a major restriction in the context of motion estimation as spatial smoothness of the motion field is required. For this reason, the domain decomposition, as it has been described since the beginning of the section, is no more used for estimating motion. The modifications that are applied for implementing the Image-based Ensemble Kalman Filter are described in the next section.

7.1.3 Smoothness property

To suppress the limitations of the localization with domain decomposition, as presented in Subsection 7.1.2, it is necessary to rethink the domain decomposition procedure. The idea is to decompose the domain with overlapping subdomains. Therefore, the domain is first decomposed, as before, in the so-called analysis subdomains D_i that span the whole image domain Ω, as schematized on Figure 7.1. Then, each analysis subdomain D_i is expanded by a given number of pixels in each direction, in order to obtain the corresponding observation subdomain E_i. This process is illustrated on Figure 7.3, where an analysis subdomain D_i is displayed in red and its corresponding observation subdomain E_i is displayed in blue.

During the analysis performed by the ensemble Kalman Filter, the Kalman gain is computed on each observation subdomain E_i independently. However, after computation of the Kalman gain on an observation subdomain E_i, only the values of $\mathbf{X}^{(a)}$ for pixels in the analysis subdomains D_i are updated. Two neighboring pixels $\mathbf{p}_1$ and $\mathbf{p}_2$, located on both sides of two neighboring analysis domains D_i and D_j, are updated by computations performed, respectively, on the two observation domains E_i and E_j. As E_i and E_j are overlapping, the Kalman gain computed for pixel $\mathbf{p}_1$ takes into account the covariance between $\mathbf{p}_1$ and the pixels of D_j, including $\mathbf{p}_2$, that belong to E_i. This ensures the smoothness in the resulting analysis at the boundaries between the analysis domains. It is possible, using the definition of the restriction operators $\mathcal{R}_D$ and $\mathcal{R}_E$ given in Subsection 7.1.1, to rewrite the Kalman analysis equation involved in this domain decomposition approach as follows:

$$\mathbf{X}^{(a),i} = \mathbf{X}^{(b),i} + \mathcal{R}_D K_E (\mathcal{R}_E \mathbf{Y}(k) - \mathbb{H} \mathcal{R}_E \mathbf{X}^{(b),i}) \tag{7.24}$$

By considering a set of overlapping subdomains E_i, the analysis obtained by the Image-based Ensemble Kalman Filter satisfies the smoothness property. However, the process leads to additional

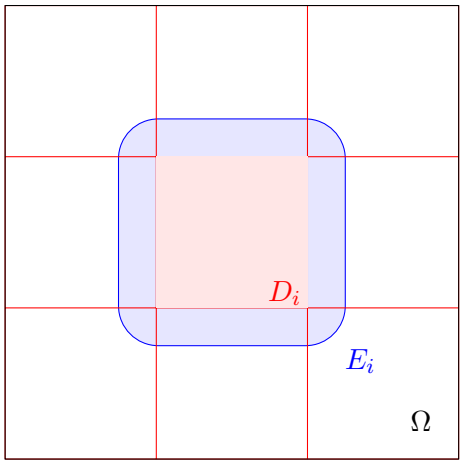

Figure 7.3 – Domain decomposition with different observation and analysis subdomains.

computations. This increase comes from the computation of Equation (7.24), as the Kalman gain is computed several times for the pixels belonging to the intersections of subdomains E_i.

Let analyze the computational cost of such implementation of localization with domain decomposition. The use of overlapping domains increases the cost proportionally to the cube of the number of pixels in the intersection. The computational cost, in case of overlapping subdomains, is equal to $O(\mathrm{N}_E{}^3)$ which is equivalent, regarding Equation (7.12), to $O((N_D + N_{Ov})^3)$. Let remind that each analysis subdomain is processed independently from the others in order to compute the analysis. Therefore K_D processors can compute the Kalman gain simultaneously, each one on an observation subdomain, and the cost becomes a function of the largest value of the number of pixels in an observation subdomain, denoted N_{EMax}. The cost is equal to $O(\mathrm{N}_{EMax}^3)$, which is equivalent to $O((\mathrm{N}_{DMax} + \mathrm{N}_{OvMax})^3)$. Considering that N_{OvMax} is small with regards to N_{DMax}, computation of the whole analysis, with the K_D subdomains, makes the global cost of the method proportional to $O(\mathrm{N}_{DMax}^3)$, if the inversion is performed in parallel. This result has to be compared with the complexity $O(\mathrm{N}_D{}^3)$ of the explicit localization method, where $\mathrm{N}_D \gg \mathrm{N}_{DMax}$. Localization with domain decomposition becomes therefore an affordable approach for estimating motion with the IEnKF.

It should be noted that limiting the number of regions is also important for keeping a low computational cost. In fact, the more regions D_i are involved in the computations, the more overlapping is needed to ensure smoothness of the estimation. If the number of overlapping pixels N_{Ovi} increases, the assumption that N_{Ovi} is small with regards to N_{Di} is no more valid. This can finally lead to prohibitory computational costs.

This section presented a domain decomposition approach enabling to localize the covariances involved in the matrices $B^{(b)}$ and R, which are used in the calculations of the Kalman gain. The localization method is based on overlapping observation subdomains. This allows satisfying the spatial smoothness property of the analysis and of its motion component. The localization with domain decomposition limits the appearance of spurious covariances between pixels linked with the small size of the ensemble. It may be applied for a small increase of the computational cost (compared to IENKF without localization) if it is used in a parallel implementation. The approach is usually applied, as described by Hunt et al. [75], with a decomposition of the spatial domain according to regular grids. The subdomains are squares or rectangles, whose size corresponds to

an empirical estimation of the decorrelation values in the horizontal and vertical directions.

However, the structures displayed on the image sequence contain information that should be considered during the assimilation process. Therefore, as described in the next section, the decomposition in subdomains should rely on structure characteristics.

7.2 Structures and domain decomposition

The explicit localization method discussed in Section 6.4 has been designed such that pixels that are distant or belonging to different structures are uncorrelated. It is possible to decompose the domain according to the two same properties. This section then describes how it is possible to decompose the domain in order to localize the covariances so that they become non negligible only for close pixels belonging to the same structures.

This proposed domain decomposition relies on a split-and-merge approach. Such segmentation technique has been widely used in image processing, in order to obtain statistically homogeneous regions. The interested Reader can, for instance, refer to the article from Horowitz and Pavlidis [77] or to the one written by Ohlander et al. [78]. The split-and-merge segmentation is based on a quadtree partition of the image. A quadtree is a tree structure in which each node has exactly four children. The split-and-merge approach starts at the root of the tree, representing the whole image. If the image is quantified as heterogeneous, it is then split into four equally sized regions. This splitting process is then iterated on each new region, and so on, as long as one region is heterogeneous. The homogeneity of a region is defined by the variance of the gray level values of pixels inside the region. It is compared with a given threshold. As long as the variance σ_i^2 of the domain D_i is above that threshold, D_i is split into four subdomains. Figure 7.4 illustrates one possible result at the end of the splitting phase.

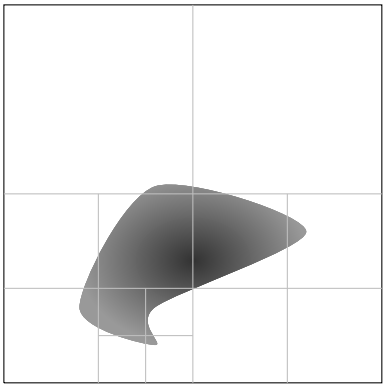

Figure 7.4 – Illustration of a domain decomposition according to the structure.

When the splitting process is finished, the initial image domain Ω is decomposed into K_D analysis subdomains D_i. An extended observation subdomain E_i is associated to each of these subdomains D_i in order to impose the smoothness property to the analysis, as explained in Subsection 7.1.3. The resulting computational cost of the approach is then equal to $O((\mathrm{N}_{DMax} + \mathrm{N}_{OvMax})^3)$, if the inversion involved in the Kalman gain Equation (7.21) is computed on K_D processors. It should be noticed that the overlapping is used to smooth the estimation over the boundaries of the analysis subdomains D_i. If the number of subdomains K_D increases the number of boundaries increases, and the whole number of overlapping pixels N_{Ov} also increases. To avoid

having too many subdomains D_i, the splitting process is then followed by a merging phase, in which regions are combined. This reduces K_D and therefore the number of overlapping pixels N_{Ov}.

When each leaf of the quadtree is considered as homogeneous, it becomes possible to merge some of them in order to obtain larger homogeneous regions. The neighboring subdomains of each leaf are analyzed and merged with it, if the homogeneity constraint keeps to be satisfied on the whole region. Such merge means that the two leaves are parts of the same structure and should be included in the same subdomain. Figure 7.5 displays an illustration of a domain decomposition oriented by structures, for which, after subdivision of the image in statistically homogeneous regions, some of them are further merged together.

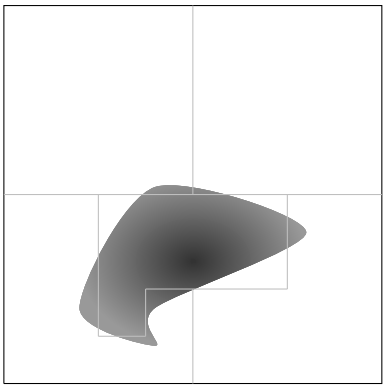

Figure 7.5 – Illustration of a domain decomposition according to the structure.

As stated before, the goal of the domain decomposition is to produce a highly parallelizable framework for localization methods. In order to include the structures in the process, the localization should be dependent from both the distance and the similarity between pixels. This remark leads to two criteria in the split-and-merge algorithm.

The first criterion defines the maximal size of an analysis subdomain D_i, as the distance between two pixels belonging to D_i should be lower than a given value. This value characterizes what "close" means in the studied application. This criterion is directly linked to the localization function ρ_d described in Section 6.4, which is depending on the distance between pixels.

The second criterion defines the notion of homogeneous region. In other words, it defines which properties should be verified by the pixels composing the analysis subdomains D_i for being considered as homogeneous. The classical choice is to consider the variance of the gray level values σ_i within the subdomain D_i. It is defined by:

$$\sigma_i = \frac{1}{\mathrm{N}_{Di}} \sum_{\mathbf{p} \in D_i} \left(I(\mathbf{p}) - \overline{I(\mathbf{p})} \right)^2 \tag{7.25}$$

where N_{Di} is the number of pixels $\mathbf{p}$ in the subdomain D_i. This criterion is directly linked to the localization function ρ_s defined in Section 6.4.

This concludes the section on structure-oriented domain decomposition. First, the domain is split into subdomains D_i following two criteria:

1. each subdomain D_i should be small enough so that covariances are only expressed for pixels which are close enough.

2. the pixels composing the analysis subdomain D_i should have similar gray values.

The whole image domain Ω being decomposed in such analysis subdomains D_i, the corresponding observation subdomains E_i are obtained by extending each D_i in every direction by a certain number of pixels, called the overlapping pixels. The domain decomposition could lead to prohibitive computational cost if the total number of overlapping pixels becomes too important. Therefore, the merging process is applied on the decomposition resulting from the splitting. Some subdomains are merged if their union does not violate the homogeneity criterion.

7.3 Conclusion

This chapter presented a localization approach enabling to reduce the impact of spurious covariances in the computation of the Kalman gain. The proposed localization approach relies on a decomposition of the discrete domain Ω into subdomains D_i. The analysis is then computed for each D_i independently. Compared to the explicit localization approach, presented in Chapter 6, the localization with domain decomposition is highly parallelizable and renders possible to estimate motion on large images in an affordable computational time. So that the estimation takes advantage of the structures displayed on the image sequence, the domain decomposition has been adapted according to these structures. The image is decomposed in regions following a split-and-merged method with two criteria: a first one defining a decorrelation distance and a second one defining the belonging or not to a particular structure.

Let recall that the decorrelation values, of the explicit localization method are defined by a function ρ. The choices made in Chapter 6 lead to a decorrelation function whose values were decreasing smoothly from 1 to 0. Therefore, every covariance value in the background error covariance matrix $B^{(b)}$ was modified by ρ, even slightly, in order to match an a priori representation of $B^{(b)}$. The localization with domain decomposition is more abrupt. The values in $B^{(b)}$ are truncated according to the decomposition in observation subdomains E_i. Some improvement should then concern combined localization methods, where both explicit localization and localization by domain decomposition are involved, as it is done by Janjic et al. in [79].

Chapter 8

Validation and applications

This chapter displays and discusses experimental results obtained by applying the method designed in Chapter 6, named Method $\mathcal{M}_4$, with explicit localization. This method is applied for estimating motion on image sequences with an Image-based Ensemble Kalman Filter, IEnKF.

The first part of the chapter, Section 8.1, is dedicated to one synthetic experiment, for which the whole knowledge of the system is available. It enables to quantify results, compute precise statistics and analyze the validity of Method $\mathcal{M}_4$. The second part, Section 8.2, is concerned with real image sequences. This allows to illustrate how the proposed method could be used in an operational framework, such as the one previously discussed in Chapter 5. As the main applications that are foreseen for the method concern the nowcasting of events in oceanography and meteorology, the developed algorithms are applied on **S**ea **S**urface **T**emperature (**SST**) images, acquired by NOAA-AVHRR sensors over Black Sea, and on Meteosat Second Generation acquisitions, displaying convective cells. However, the method may be applied on other applicative area, outside of these domains. Consequently some additional results are given on traffic sequences displaying cars.

8.1 Synthetic Experiment

Method $\mathcal{M}_4$ presented in Chapter 6, which involves an explicit localization, is first tested on a synthetic twin experiment.

Starting from the initial image and motion field, displayed on Figure 8.1, the Image Model $\mathbb{M}$ is integrated in time, as explained in Subsection 6.2.1 of Chapter 6. In such context, the motion field $\mathbf{w}$ is advected by itself and the synthetic image I is transported by the motion field $\mathbf{w}$.

The temporal integration of the Image Model produces a discrete sequence of motion fields and a sequence of snapshots. The motion fields obtained from that integration will be further used as ground truth for evaluating results and computing statistics. Six snapshots are taken every ten time steps. They are used as image observations for the assimilation experiment. Three snapshots of the sequence are displayed on Figure 8.2, simultaneously with the ground truth on motion at the same dates.

The observations ensembles are constructed from the snapshots, as explained in Section 6.3. The process producing the snapshots is considered as noisy and the observations ensemble should statistically represent this uncertainty. The noise is considered to follow a zero-mean Gaussian law with covariances given by the matrix R_t. R_t is supposed diagonal, therefore the noise is uncorrelated in space with a standard deviation taken as representing 10% of the acquired value for each pixel.

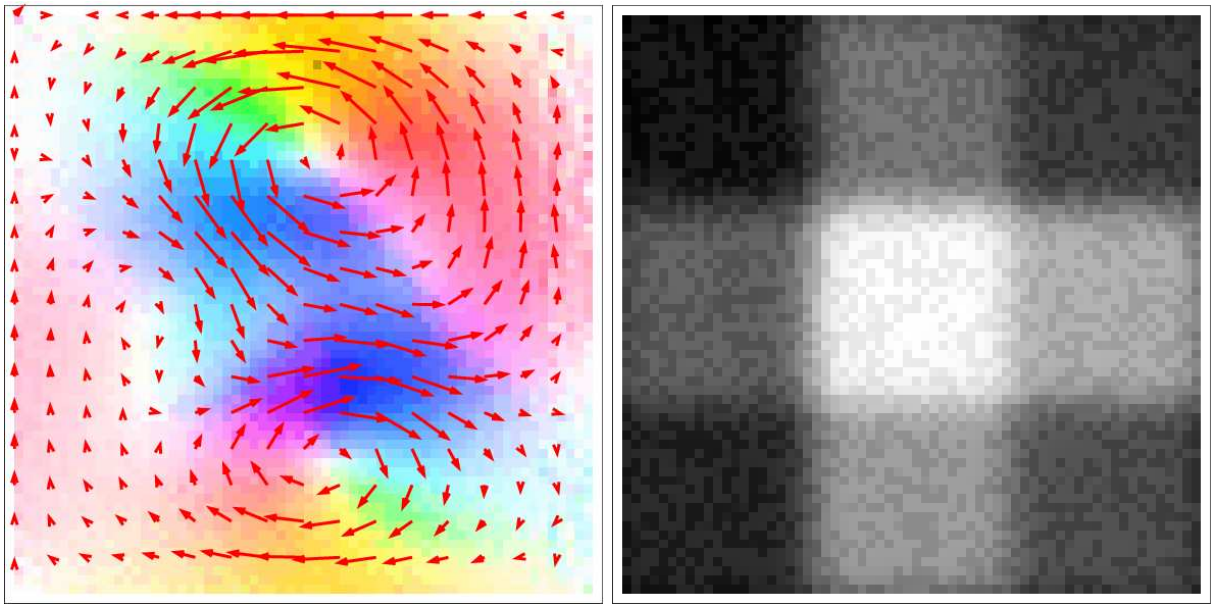

Figure 8.1 – Initial conditions of ground truth.

For each observation image I_t^O, obtained at date t, an observations ensemble of 50 members is generated. The value of the member indexed by i, $\mathbf{Y}_t^i$, is defined according to:

$$\mathbf{Y}_t^i = I_t^O + \mathcal{E}_{R_t} \tag{8.1}$$

where $\mathcal{E}_{R_t}$ is the result of a random draw according to the Gaussian law $\mathcal{N}(0, R_t)$.

On another hand, an ensemble of 50 motion members is created as explained in Subsection 6.2.2. These motion fields are the results of state-of-the-art optical flow algorithms applied on the first two snapshots. This process ensures to obtain a good estimation of the background error covariance matrix $B_0^{(b)}$. In fact, if every optical flow algorithm estimates the same motion vector at pixel $\mathbf{p}$, the resulting $B_0^{(b)}$ computed from the ensemble and given by:

$$B_0^{(b)} \approx \overline{(\mathbf{X}_0^{(b),i} - \mathbf{X}_0^{(b)})(\mathbf{X}_0^{(b),i} t - \mathbf{X}_0^{(b)})^T} \tag{8.2}$$

will have a low variance value on the motion component for that pixel $\mathbf{p}$. If the state-of-the-art methods estimate a broad range of different motion vectors on a given pixel $\mathbf{p}$, the resulting variance value on the motion component in $B_0^{(b)}$ will be high. The thereby generated ensemble is representative of the uncertainty on the motion field at initial date. The Image-based Ensemble Kalman Filter will then be used to better motion on the pixels for which the uncertainty is high.

The Image-based Ensemble Kalman Filter is applied to estimate motion either without localization, using Equations:

$$\mathbf{X}_t^{(a),j} = \mathbf{X}_t^{(b),j} + K(\mathbf{Y}_t - \mathbb{H}\mathbf{X}_t^{(b),j}) \tag{8.3}$$

with:

$$K = B_t^{(b)}\mathbb{H}^T(\mathbb{H}B_t^{(b)}\mathbb{H}^T + R_t)^{-1} \tag{8.4}$$

or with Method $\mathcal{M}_4$ relying on an explicit localization, as described in Chapter 6 by:

$$\mathbf{X}_t^{(a),i} = \mathbf{X}_t^{(b),i} + K_L(\mathbf{Y}(k) - \mathbb{H}\mathbf{X}_t^{(b),i}) \tag{8.5}$$

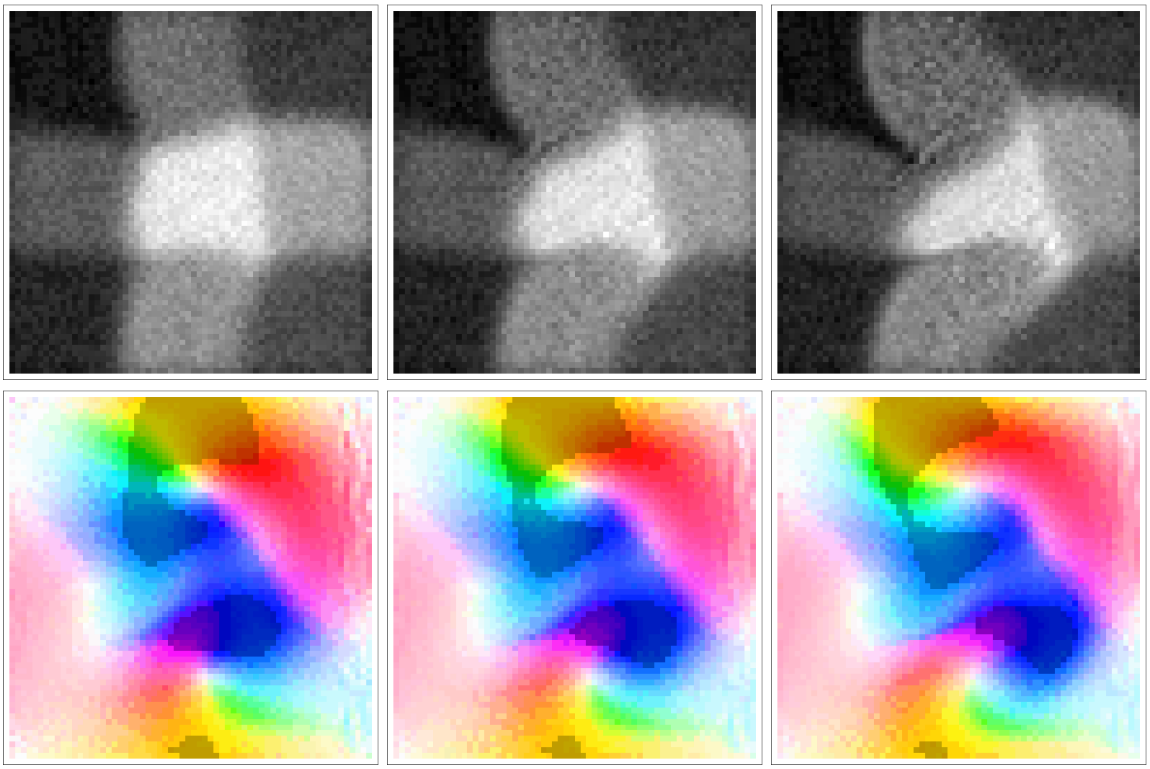

Figure 8.2 – Snapshots of the synthetic experiment (*top line*). Motion fields at the same dates (*bottom line*).

where:

$$L_t^{(b)} = \rho \circ B_t^{(b)} \tag{8.6}$$

and:

$$K_L = L_t^{(b)} \mathbb{H}^T \left(\mathbb{H} L_t^{(b)} \mathbb{H}^T + R_t \right)^{-1} \tag{8.7}$$

In case of explicit localization, the decorrelation distance that is used for defining the localization function ρ_d has a value of 1. The localization function ρ_s, relying on the similarity between pixels, gets a decorrelation parameter value of 0.1.

Errors statistics on the discrepancy between estimation and ground truth are computed. Those concerning the absolute value of angular errors are illustrated on Figure 8.3. Let denote $\mathcal{E}_\theta^i(t)$ the error computed with the motion field $\mathbf{w}_t^i$ of member $\mathbf{X}_t^i$. $\mathcal{E}_\theta^i(t)$ is defined as the average of the angular error and is computed by:

$$\mathcal{E}_\theta^i(t) = \frac{1}{\mathrm{N}_\Omega} \sum_{\mathbf{p} \in \Omega} \left| \theta_{GT}(\mathbf{p}, t) - \theta^i(\mathbf{p}, t) \right| \tag{8.8}$$

where $\theta(\mathbf{p}, t)$ is the angle formed by $\mathbf{w}(\mathbf{p}, t)$ and a reference direction defined by $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$. N_Ω denotes the number of pixels in the discrete domain Ω. This criteria, applied to each member, allows to visualize the individual errors of members at a given date t as a scalar. The values of the angular errors $\mathcal{E}_\theta^i(t)$ of 20 members (randomly chosen among the set of 50) are visualized, on Figure 8.3, as

blue points, at each date of the studied temporal interval. The error $\mathcal{E}_\theta(t)$ of the ensemble mean $\mathbf{w} = \overline{\mathbf{w}^j}$ (mean over the 50 members) is computed by:

$$\mathcal{E}_\theta(t) = \frac{1}{\mathrm{N}_\Omega} \sum_{\mathbf{p}\in\Omega} |\theta_{GT}(\mathbf{p},t) - \theta_{\mathcal{M}_4}(\mathbf{p},t)| \tag{8.9}$$

and represented by the red curve.

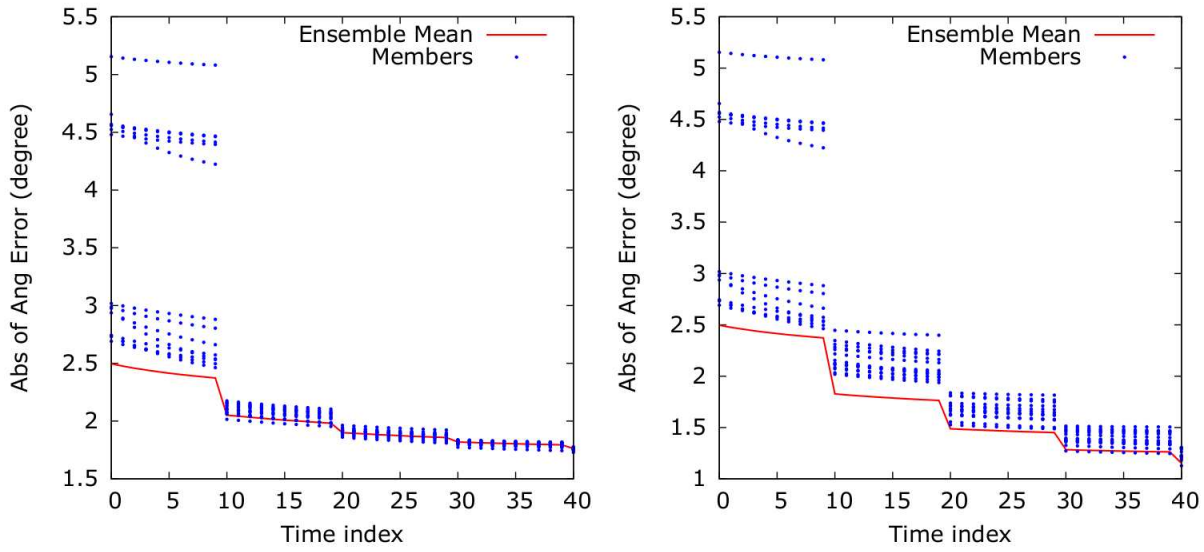

Figure 8.3 – Absolute values of the angular error without localization (*left*) and with localization (*right*). The blue points correspond to one member of the ensemble and the red curve to the ensemble mean.

As it can be seen on Figure 8.3, the initial ensemble is widely spread and the angular error $\mathcal{E}^i_\theta(0)$ among the members goes from 2.65 to 2.15 degrees. The first analysis step, at date 10, reduces the errors of approximately 30 percent as, for the ensemble mean, $\mathcal{E}_\theta(10)$ equals 2.4 whereas $\mathcal{E}_\theta(11)$ equals 1.8.

One can see the effects of the localization when comparing the left figure, statistics without localization, with the right one, statistics with localization.

- First effect: the ensemble is spread enough for a longer duration. Keeping the ensemble spread large enough, as long as possible, is of major interest for online image processing for the following reason. When the shrinking is important, the background error covariance matrix $B_t^{(b)}$ gets small variance values. At the limit, when the matrix is almost null, the analysis step has no more effect on the ensemble. It is clear that if $B_t^{(b)}$ is null so is the Kalman gain in Equation (8.4), and so is the innovation added in Equation (8.3).

- Second effect: as visible at the first analysis step, the localization increases the accuracy of the estimation. At the end of the studied temporal interval, the angular error, in degrees, is 0.7 with localization and 1.7 without.

- Third effect: the accuracy of the estimation still improves at the 5^{th} or 6^{th} analysis step with localization, whereas no effect occurs anymore after the 4^{th} computation of an analysis without localization.

Figures on the relative norm errors are similar and displayed on Figure 8.4.

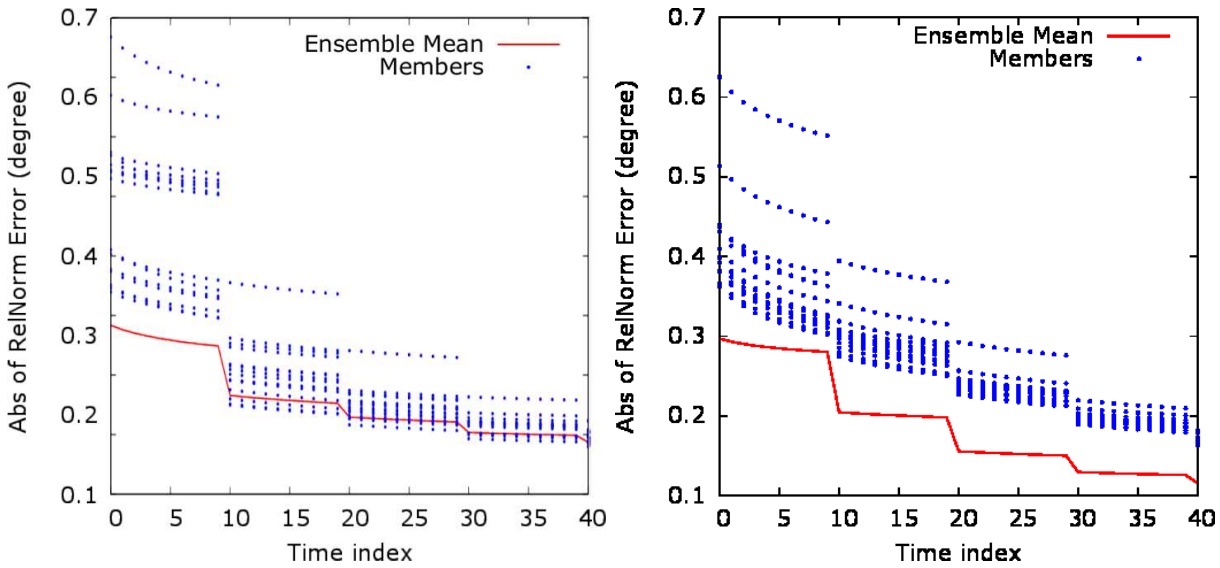

Figure 8.4 – Values of the relative norm error without localization (*left*) and with localization (*right*).

Figure 8.5 displays, on the left, the norm of the error between the estimation and the ground truth at the end of the temporal interval. The variance of the ensemble at the same date is displayed on the right. It can be observed that the spread of the ensemble has a good geographical correspondence with the errors. At the end of the studied temporal interval, the variance of the ensemble gives an estimation of the uncertainty of the estimation. Regions where the uncertainty is high actually correspond to regions where the error on the estimation at the end of the temporal interval is high. This means that the method correctly localizes the regions where the estimation is uncertain. It could be used, for instance, to redefine a new ensemble for further assimilation. This correct localization of the uncertainty is a valuable information for any decision process relying on an assimilation experiment. This makes this implementation of the Image-base Ensemble Kalman filter appropriate for operational use.

After computing motion between the 5^{th} and 6^{th} observations with the whole set of motion estimation algorithms (see Subsection 6.2.2 of Chapter 6) and quantifying their results, the best method shows an average angular error of 5 degrees. This has to be compared with the 1-degree error obtained by our Image-base Ensemble Kalman Filter with explicit localization. Figure 8.6 displays from left to right, the ground truth, the estimation obtained with the IEnKF and the best result of the optical flow algorithms.

8.2 Experiment on Real Data

In this section, all results have been obtained using an implementation of the Image-based Ensemble Kalman Filter with an explicit localization, as the importance of the localization process has been proved in Section 8.1.

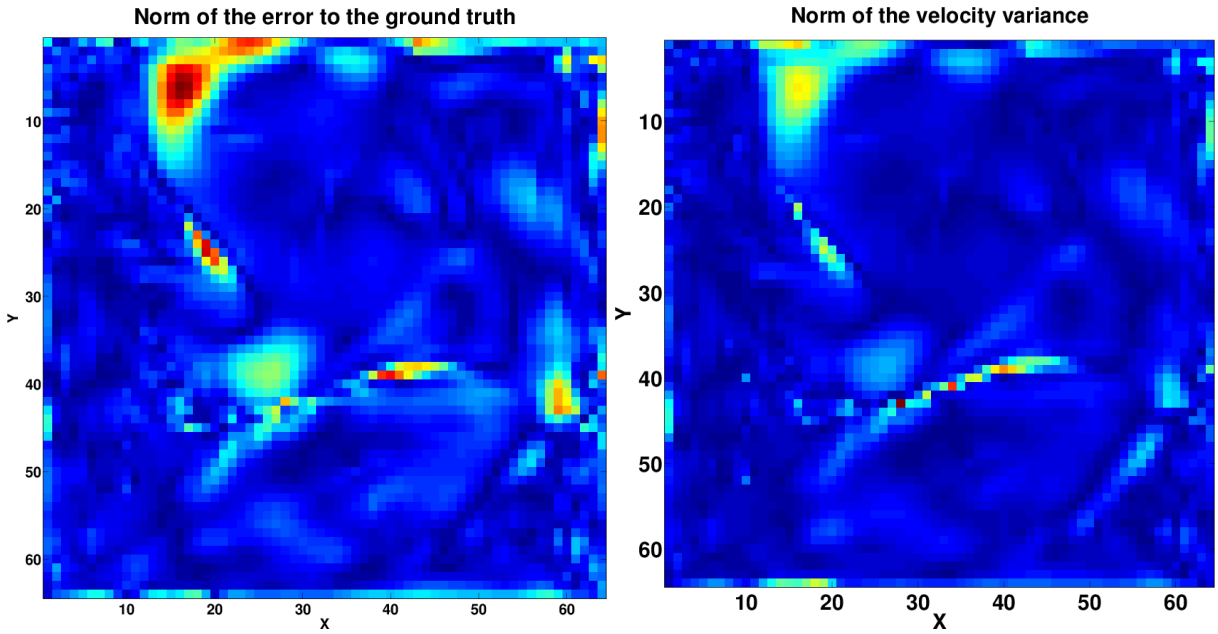

Figure 8.5 – *Left*: Norm of the error between the estimation and the ground truth. *Right*: Variance of the ensemble results.

8.2.1 Motion estimation on meteorological sequences

IEnKF was applied on meteorological satellite data. Experiments were done on sequences displaying convective cells, as these structures are clearly visible on the image data and may be easily tracked. The foreseen operational application is to improve the short term forecast of cloud cover. A sequential filter allows to estimate the uncertainty of the estimation and is highly parallelizable. Its use in an operational context would then be particularly interesting to estimate the accuracy of the cloud cover forecast and to reduce the computational time required for processing large size satellite images.

The images of the first studied meteorological sequence are acquired every 15 minutes, in the infrared domain, with a 5 kilometer resolution, by the Meteosat Second Generation (MSG2) satellite.

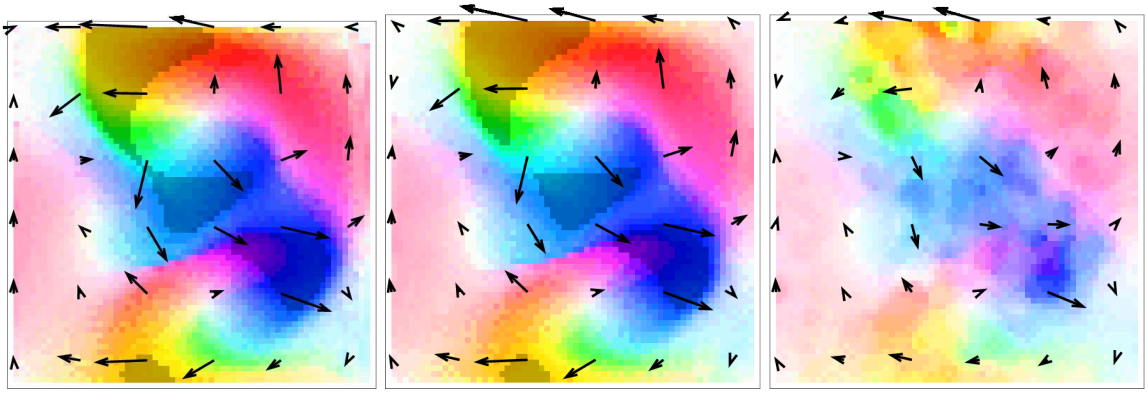

Figure 8.6 – From left to right. Ground truth; IEnKF; best result of optical flow methods.

Results are given on Figure 8.7 with the display of the boundary of one cloud, which is advected by the estimated motion field. One can see on the images that the green curve stays on the boundary

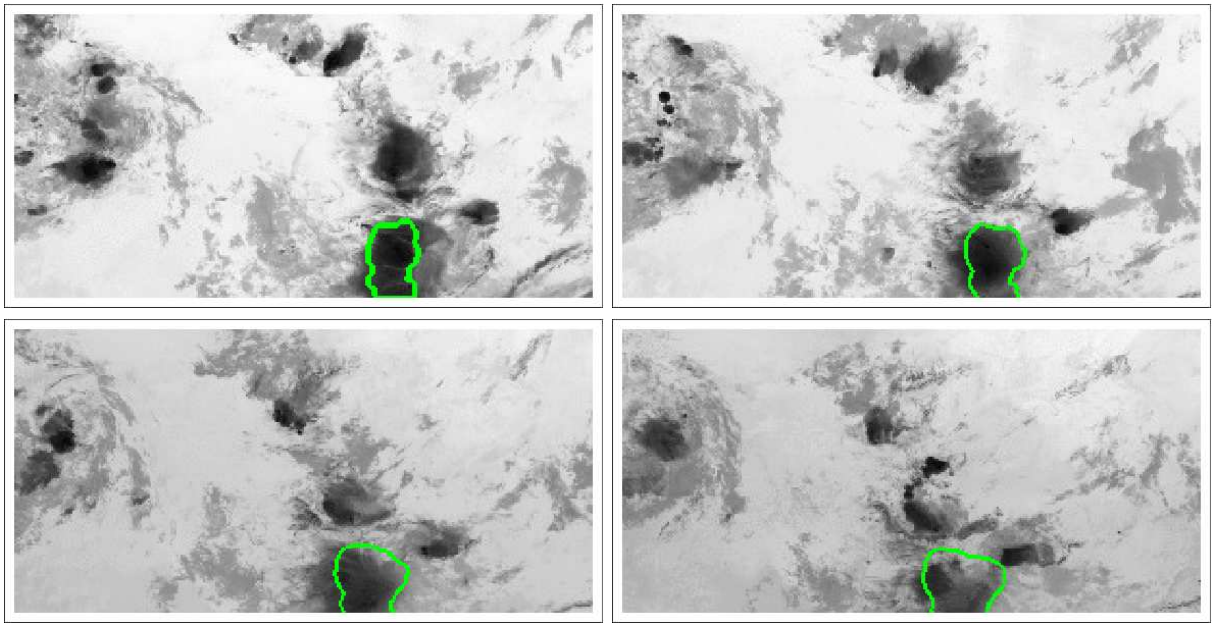

Figure 8.7 – Advection of the boundary by the estimated motion field.

of the convective cell during the time integration. This accurate tracking confirms that the cell motion is correctly estimated.

A second image sequence from the MSG2 satellite has been studied and results are given on Figure 8.8 by advecting the boundary of one cloud with the estimated motion field. This again demonstrates correctness of the estimation, as the red curve stays on the boundary of the chosen convective cell.

8.2.2 Motion estimation on traffic sequences

In order to demonstrate the broad variety of possible applications of IEnKF, experiments were done on traffic images from the database KOGS/IAKS of the Karlsruhe University [80]. Four images of the first sequence are given on Figure 8.9. On this sequence, a bright taxi is turning to the right. This sequence is particularly interesting as a black car moves from the left to the right of the image, while a dark van moves in the opposite direction from the right. These two vehicles are hard to distinguish from the background of the image and lead to usually high estimation errors. The implementation of IEnKF, described in Chapter 6, is used in order to estimate motion of all cars. As a quality assessment is not at hand with the arrow or color displays of motion, results are visualized on Figure 8.10, thanks to trajectories of characteristic points. Each trajectory is obtained by advecting the chosen characteristic points with the estimated motion field. The advection is computed according to Equation (6.41) in Subsection 6.2.1, which implies that the position of the characteristic point can be obtained with any chosen temporal resolution. Let remind that this

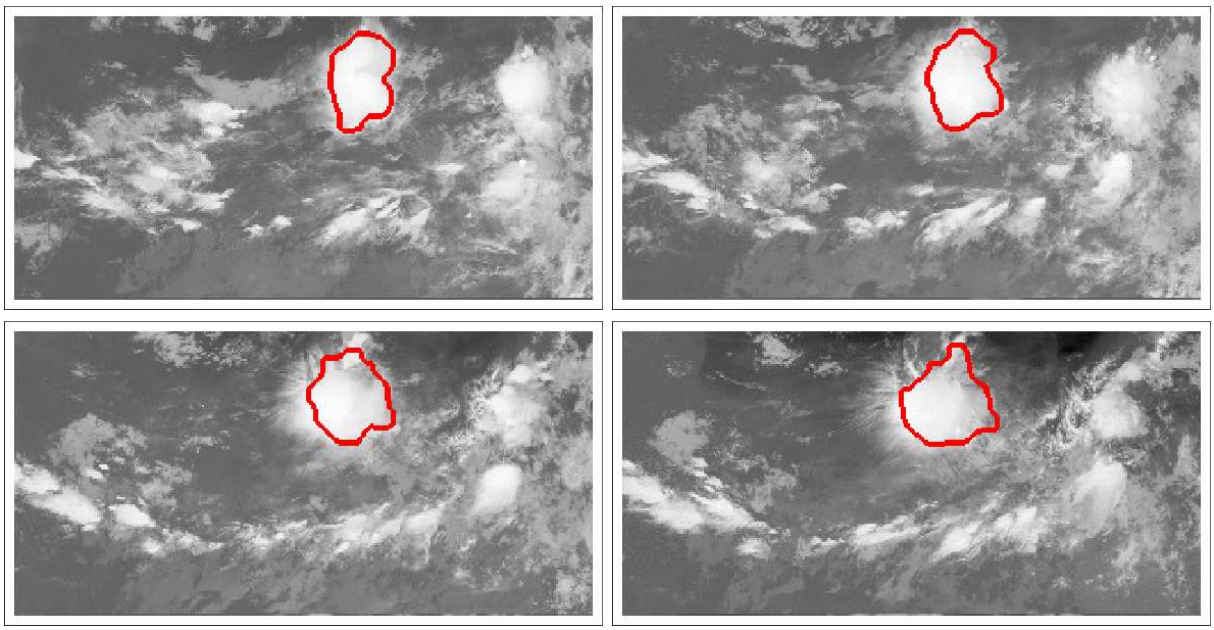

Figure 8.8 – Advection of the boundary by the estimated motion fields.

equation is the optical flow, given by:

$$\frac{\partial I}{\partial t} = -\mathbf{w} \cdot \nabla I \tag{8.10}$$

On each image of the figure, a green colored circle displays the current position of one characteristic point and the corresponding green curve describes its whole trajectory. As it can be seen, motion is correctly estimated, in norm and orientation, for the white taxi and for the dark van coming from the right. This conclusion is still valid when defining new characteristic points. However, the black car arriving from the left gets weaker quality results due to its low contrast with the background

Four images of a second sequence of the same database are shown on Figure 8.11. This sequence displays vehicles driving after a traffic light turned green. The vehicles accelerate and estimation becomes challenging.

This sequence has been processed with the implementation of the Image-based Ensemble Kalman Filter given in Chapter 6. Results are given on Figure 8.12. It displays trajectories of the characteristic points on the first and last images with green, blue and red points.

As cars are accelerating when the traffic light turns green, the algorithm slightly underestimates their speed. A future implementation of the IEnKF should involve an acceleration term in the state vector, as described by Béréziat et al. [41] in order to improve the estimation.

Four images of the third sequence are displayed on Figure 8.13. Again a traffic light turned green and cars accelerate. Moreover, one car undertakes a U-turn.

Results obtained by IEnKF are displayed on Figure 8.14. The computed trajectories of three characteristic points are shown in green, red and blue. One can visualize the car undertaking a U-turn.

Figure 8.9 – First traffic sequence.

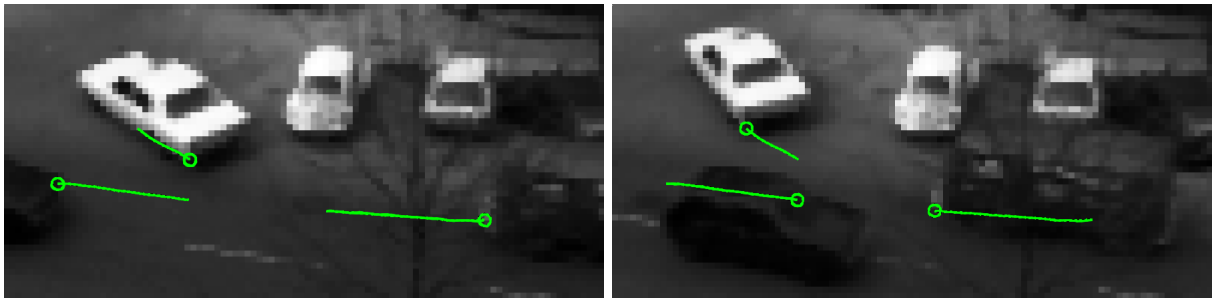

Figure 8.10 – Motion results given as trajectories, with the position (*circle)* of a characteristic point on the first frame (*left)* and the last one (*right).*

8.3 Conclusion

This chapter first illustrated, using a synthetic experiment, why localization is necessary for IEnKF, in order to obtain a valuable motion estimation from an image sequence. Second, it demonstrated the operational possibilities of the approach through real cases experiments. On meteorological satellite acquisitions, the method demonstrates its performance with the correct tracking of clouds boundary. On traffic images, the approach allows a correct estimation of motion, which is demonstrated from the computation of trajectories of characteristic points on moving cars.

Figure 8.11 – Second traffic sequence.

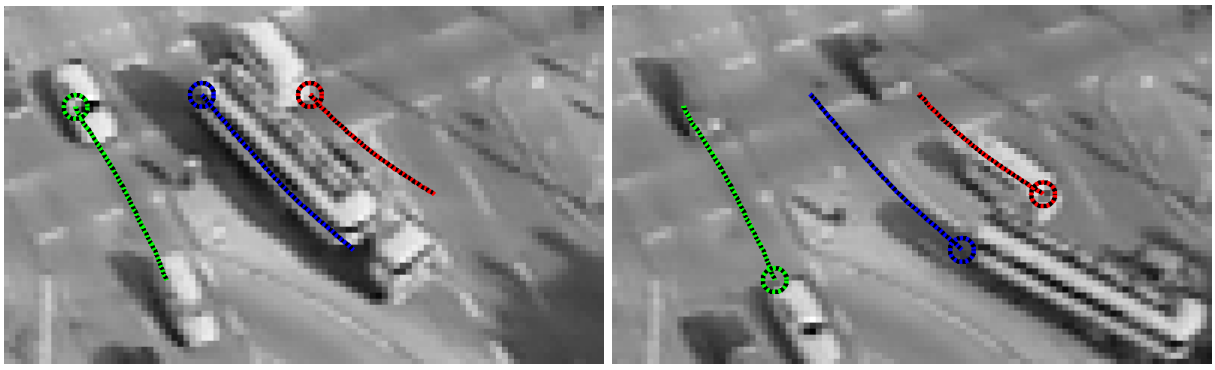

Figure 8.12 – Motion results given as trajectories, with the position (*circle)* of a characteristic point on the first frame (*left)* and the last one (*right)*.

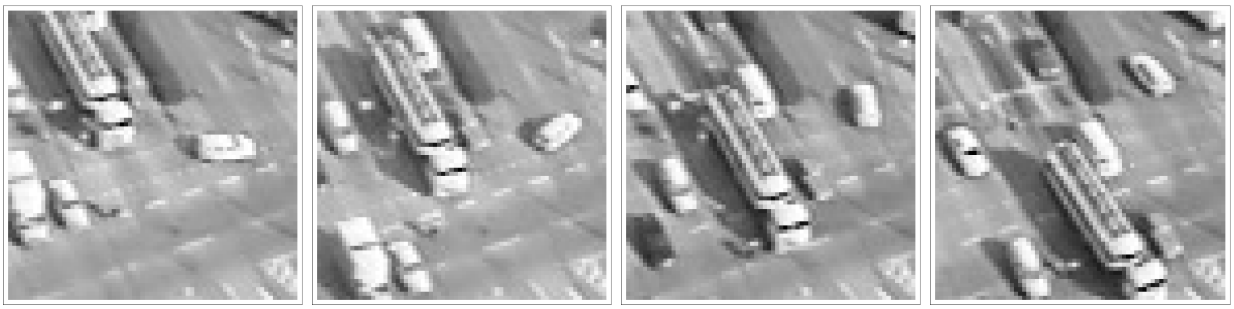

Figure 8.13 – Third traffic sequence.

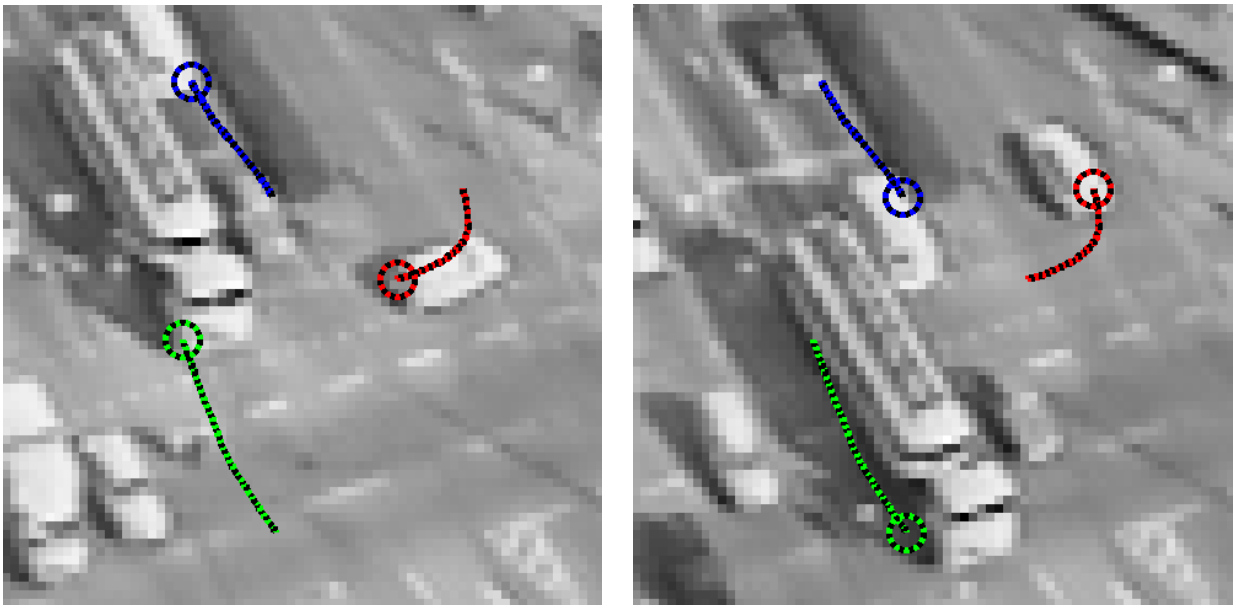

Figure 8.14 – Motion results given as trajectories with the position (*circle)* of a characteristic point on the first frame (*left)* or the last one (*right)*.

Conclusion

Data assimilation methods have been used during the last decade in the image processing community. Béréziat et al. [1] discuss for instance the problem of motion estimation by an incremental method and the use of error covariance matrices to filter noisy or occluded data. Sequential methods, such as the Kalman Filter have also been used, for instance by Elad et al. [2], for motion estimation. This thesis has investigated both of these methods: Part I focused on variational image assimilation and Part II treated sequential filters.

It is possible to consider the structures, which are displayed in the image sequence, in both types of estimation method. As an example, Papadakis et al. [3] describe a variational assimilation method where motion is estimated by using the segmentation of the structures visualized on the sequence. Modeling the structures in the data assimilation process is equivalent with correlating in space the errors to be minimized by the estimation method. This leads to better estimation results. This thesis focuses on how information about structures could be added in the estimation process.

Variationnal Assimilation

The variational data assimilation algorithm used in the document, called 4D-Var, relies on three equations: an evolution equation, describing the physical knowledge on the studied system; a background equation, representing the a priori knowledge on the system state; and an observation equation, linking the state vector and the observations, which are, in our applications, computed from image acquisitions. The problem of motion estimation is solved by minimizing the errors associated to the background and to the observation equations. The mathematical description of such an approach has been given in Chapter 1.

Improvements of this initial approach are possible. For instance, a major improvement is to add an error term, which expresses the uncertainty on the evolution equation, in the cost function to be minimized. Such a method is named weak-constraint 4D-Var. Weak 4D-Var methods have shown their ability to estimate the acceleration terms or the external forces that are not included in the evolution laws, as described by the paper from Béréziat et al. [41]. It should then be possible to enhance the quality of the forecasts and estimations given in this thesis document by considering a term representing the model error.

However, the mathematical descriptions given in this thesis focus on modeling the structures and defining their evolution laws, as it is the core subject of the research work.

A first approach, described in Chapter 2, consisted in designing a method whose inputs are low level data: the acquired image sequence and the contour points computed on these images. The structures are associated to their boundary, whose topology can change over time. Therefore, considering multiple structures, and merging or splitting some of them, is straightforward. The choice

has been made to describe these structures by an implicit function, whose null value corresponds to the boundary curve. Each other value at a given pixel is its signed distance to the closest boundary point. The Image Model $\mathbb{M}$ characterizes the evolution of the images and describes the structures' displacement. Then, the estimation process tries to match the boundary of the structures, as given by the state vector $\mathbf{X}$, to the contour points computed on the image acquisitions. Therefore, the Image Model must be defined accordingly to the structures representation.

A possible perspective of this research work would be to investigate different structures coding, depending on the application. When quantifying cars displacement, it is not necessary to allow the topology of the boundary to change over time, as a car can always be represented by a quadrangle. Similarly, hurricanes can be represented by the position of their eye and a radius parameter. Considering alternative representations would however necessitate to design their corresponding evolution model.

During the estimation process, the background error, associated to the background equation, and the observation error, associated to the observation equation, are minimized. These error terms are supposed Gaussian and zero-mean and are therefore modeled by their corresponding covariance matrices: the background error covariance matrix and the observation error covariance matrix. In the methods described in Chapter 1 and Chapter 2, those matrices have been designed as diagonal, i.e. without any covariance between locations nor variables.

However, as the signed distance map, used in Chapter 2, imposes constraints between pixels, the minimum is achieved while considering spatial correlations of the motion vectors on the image domain. Therefore, Chapter 3 describes an approach for modeling the covariances of the background error for the motion field. To mimic the impact of the distance map on the motion result, the covariances are modeled according to a function of the distance to the boundary. Although, modeling the background error covariances is strongly beneficial for the estimation, the size of the covariance matrix renders its storage unaffordable. Another solution is needed in order to consider these covariances. The choice has been made to model the background error covariances of motion as a local function approximating the Laplacian of motion. This allows to model the covariances at a quasi null storage cost and additional computational cost, the local function only considering a few neighboring pixels.

Perspectives on this work should concern the covariances between the motion and the image variables of the state vector. Moreover a model of the observation error covariances should also be considered. To do so, alternatives of the literature, which model the covariance matrices at low computational cost, should be investigated. For instance, Chabot et al. [49] describe an approach where the covariance matrices are represented in a space of low dimension, which solves the storage issue.

The last contribution on variational image assimilation, described in the document, consists in developing a nowcasting algorithm for precipitation forecast. It is discussed in Chapter 5. This work has been done in collaboration with the company Weather Measures and is based on radar acquisitions. The radar measures the reflectivity, which is further transformed into a precipitation rate. This allows, for instance, to evaluate the future rain quantity at a given location. State-of-the-art algorithms for precipitation nowcasting rely on two components: first, an estimation phase, based on several consecutive acquisitions, estimates the wind field; second, a forecast phase uses the estimation and the last acquisition for generating future precipitation rates.

State-of-the-art algorithms mainly differ on their estimation method. Most of them only rely on image acquisitions and structures segmentation. Moreover, they often make use of characteristic

features such as the rain cells segmentation and labeling on each acquisition, which highly constrain the estimation method. Using the approach described in Chapter 2, our method tends to be more general. First, cells of high precipitation rates are obtained by applying a threshold on the acquisition. Then, without any labeling of the cells, the estimation method is able to estimate their displacement and growth rate.

The proposed forecast phase relies on a more realistic physical model and allows prevision at any temporal horizon. The approach, as described in the document, shows satisfying results and good forecast skills in an operational context.

On this subject, the perspectives mostly concern the operational setting. The current system uses measures from a single radar. It is however well known that radar images suffer from lots of acquisition problems. Using a network of radars could allow to estimate a reliability factor for each acquisition and thus drastically enhance the resulting forecasts. Moreover, the city of Clermont-Ferrand possesses a network of rain gauges, whose measures could also be used to calibrate the radar acquisitions.

Sequential assimilation

The main drawback associated to the variational methods is their computational time. Thus alternative methods, based on sequential filtering, have been investigated in Part II of this thesis document.

The approach described in Chapter 6, based on the **En**semble **K**alman **F**ilter (**EnKF**), produces an estimation of motion as well as the uncertainty on this estimation by using: an Image Model; a background value and the probability density function of its a priori error; an observation, an observation operator and the PDF associated to the observation error. Both probability density functions are supposed Gaussian and zero-mean. Therefore each PDF is fully defined by its corresponding covariance matrix, respectively called the background error covariance matrix and the observation error covariance matrix. As these matrices can not be stored in memory, when considering an application on images, they are sampled, respectively, through an ensemble of state vectors and an ensemble of observation vectors.

To obtain an accurate estimation, the state vectors ensemble at initial date must correctly sample the uncertainty on the background value.

In the proposed methods, the design of the initial ensemble is based on the knowledge that a broad bunch of optical flow algorithms is available in the literature of image processing and computer vision. They differ in their heuristics, their formulation and their parametrization. Each of these optical flow algorithms is adapted to different image contexts. By applying a wide bunch of them in-between the first two frames of the image sequence, an ensemble of motion fields is created. This ensemble is then used to estimate both the motion background at initial date and its uncertainty.

Each member of the state vectors ensemble is then integrated in time by the Image Model until the first acquisition date.

The members of the state vectors ensemble, at that acquisition date, are used to approximate the background error by computing the associated background covariance matrix. Similarly, an ensemble of observations is created, according to image properties, in order to sample the observation error matrix. An estimation of the state of the system can then be computed by using the EnKF equations, the output being an ensemble of analysis state vectors. The mean of the analysis

state vectors furnishes the estimation, while the disparity of the ensemble, through the analysis covariance matrix, estimates the error on the estimation.

However, as summarized in the following, using the ensemble Kalman filter on real observations implies to consider correction techniques.

In our application, an ensemble of observations is necessary to avoid the under-estimation of the analysis covariance matrix. However, to get rid of the ensemble of observations without under-estimating the analysis covariance matrix, the method presented by Bishop et al. [81] called the **E**nsemble **T**ransform **K**alman **F**ilter (**ETKF**) should have been investigated. The ETKF computes the exact value of the analysis covariance matrix and, then, deduces the ensemble of analysis state vectors that best samples this matrix. This resampling stage is purely computational and relies on a singular value decomposition. Therefore, the resulting members may take physically absurd values. Applying this ETKF algorithm for motion estimation would imply to constrain the resampling stage to physically significant members.

A major drawback of the EnKF comes from the approximation of the background error covariance matrix by the ensemble members. Sampling this matrix makes spurious covariances appear, for instance between distant pixels. This drawback imposes to use localization methods. Distance dependent localization methods have been widely used, as described by Hamil et al. [64] or by Miyoshi et al. in [82], and are based on the hypothesis that distant pixels should be less correlated than closer ones. A new localization criteria has been designed in Chapter 6, that considers both the distance and the similarity between pixels. This allows close pixels belonging to different structures to be less correlated than pixels more distant but belonging to the same structure. A localization method, called explicit localization, is then applied by multiplying the background error covariance value with the value of the localization function.

However, the explicit localization can not be used for large images. This approach increases the rank of the background covariance matrix, which can no more be stored in memory because of its size. Moreover, the computational time necessary for the inversion needed in the Kalman gain becomes prohibitive.

Another localization approach, described in Chapter 7, has been investigated. It is based on a decomposition of the image domain into subdomains, on which the analysis are computed independently. This enables to localize the covariances while keeping the computational cost affordable. This approach takes the structures into account, while decomposing the domain into subdomains. For that, it mimics the localization function used by the explicit localization.

However, localization with domain decomposition, even if the subdomains overlap on their neighbors, leads to some discontinuities at the domains boundary. In fact, the covariance matrix resulting from this localization is only a truncated version of the original one. The main perspective on the localization with domain decomposition would be to combine it with an explicit localization. Such techniques, combining explicit localization and localization with domain decomposition, have already been developed, for instance by Janjic et al. [79]. Considering the domain decomposition oriented by structures and the localization function relying on similarity between pixels should lead to estimations without discontinuities and a computationally efficient algorithm.

Bibliography

[1] D. Béréziat and I. Herlin, "Solving ill-posed image processing problems using data assimilation," *Numerical Algorithms*, vol. 56, no. 2, pp. 219–252, February 2011. 3, 13, 141

[2] M. Elad and A. Feuer, "Recursive optical flow estimation - adaptive filtering approach," *Journal of Visual Communication and Image Representation*, vol. 9, pp. 119–138, 1996. 3, 141

[3] N. Papadakis and E. Mémin, "A variational technique for time consistent tracking of curves and motion," *Journal of Mathematical Imaging and Vision*, 2008. 3, 141

[4] G. Evensen, "The ensemble Kalman filter: Theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, pp. 343–367, 2003. 5, 13, 97

[5] B.K.P. Horn and B.G. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981. 11, 107

[6] J. Hadamard, *Lecture on Cauchy's Problem in Linear Partial Differential Equations*, Yale University Press, New Haven, 1923. 11, 32

[7] A. N. Tikhonov, "Regularization of incorrectly posed problems," *Soviet Mathematics - Doklady*, vol. 4, pp. 1624–1627, 1963. 11

[8] H. H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *Pattern Analysis and Machine Intelligence*, vol. 8, no. 5, pp. 565–593, September 1986. 12

[9] M. Nielsen, L. Florack, and R. Deriche, "Regularisation and scale space," Tech. Rep. RR 2352, INRIA, Septembre 1994. 12

[10] M. Werlberger, T. Pock, and H. Bischof, "Motion estimation with non-local total variation regularization," in *Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, June 2010. 12

[11] D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *European Conference on Computer Vision*, 2010, pp. 2432–2439. 12, 32, 72, 107, 108

[12] D. Fortun, P. Bouthemy, and C. Kervrann, "Optical flow modeling and computation: A survey," *Computer Vision and Image Understanding*, vol. 134, pp. 1 – 21, 2015, Image Understanding for Real-world Distributed Video Networks. 12, 107

[13] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal on Computer Vision*, vol. 92, no. 1, pp. 1–31, March 2011. 12, 32, 65, 72, 107

[14] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conference on Computer Vision*. October 2012, Part IV, LNCS 7577, pp. 611–625, Springer-Verlag. 12

[15] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer, "Modeling temporal coherence for optical flow," in *International Conference on Computer Vision*. IEEE, 2011, pp. 1116–1123. 12

[16] C. Vogel, S. Roth, and K. Schindler, "View-consistent 3D scene flow estimation over multiple frames," in *European Conference on Computer Vision*, pp. 263–278. Springer, 2014. 12

[17] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4 2006, pp. 13, 2006. 12

[18] A. Smeulder, D. Chu, R. Cucchiara, S. Calderara, A. Deghan, and M. Shah, "Visual tracking: an experimental survey," *Pattern Analysis and Machine Intelligence*, July 2014. 12

[19] N. Peterfreund, "Robust tracking of position and velocity with Kalman snakes," *Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, June 1999. 12

[20] Y. Rathi, N. Vaswani, A. Tannenbaum, and A.J. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, August 2007. 12

[21] C. Avenel, E. Mémin, and P. Pérez, "Tracking closed curves with non-linear stochastic filters.," in *Conference on Space-Scale and Variational Methods*, 2009. 12

[22] F. Bouttier and P. Courtier, "Data assimilation concepts and methods," Tech. Rep., Training Course of European Centre for Medium-Range Weather Forecasts, 1999. 13

[23] N. Papadakis, E. Mémin, A. Cuzol, and N. Gengembre, "Data assimilation with the weighted ensemble Kalman filter," *Tellus Series A : Dynamic meteorology and oceanography*, vol. 62, no. 5, pp. 673–697, October 2010. 13

[24] M. Ridal, M. Lindskog, N. Gustafsson, and G. Haase, "Optimized advection of radar reflectivities," *Atmospheric Research*, vol. 100, no. 2–3, pp. 213–225, 2011, Uncertainty Propagation in Advanced Hydro-Meteorological Forecast Systems. 13, 91

[25] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960. 13, 17, 97, 102

[26] F. Le Dimet and O. Talagrand, "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects.," *Tellus Series A : Dynamic meteorology and oceanography*, pp. 97–110, 1986. 13, 29, 48

[27] R. LeVeque, *Numerical Methods for Conservative Laws*, Lectures in Mathematics. ETH Zürich, Birkhaüser Verlag, 2nd edition, 1992. 25

[28] A. Robert, "A stable numerical integration scheme for the primitive meteorological equations," *Atmosphere-Ocean*, vol. 19, no. 1, pp. 35–46, 1981. 27

[29] A. Staniforth and J. Côté, "Semi-Lagrangian integration schemes for atmospheric models - A review," *Monthly Weather Review*, vol. 119, no. 9, pp. 2206–2223, 1991. 27, 88

[30] J Pudykiewicz and A Staniforth, "Some properties and comparative performance of the semi-Lagrangian method of Robert in the solution of the advection-diffusion equation," *Atmosphere-Ocean*, vol. 22, no. 3, pp. 283–308, 1984. 27

[31] R. H. Byrd, P. Lu, and J. Nocedal, "A limited memory algorithm for bound constrained optimization," *Journal on Scientific and Statistical Computing*, vol. 16, no. 5, pp. 1190–1208, 1995. 30

[32] 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. 30, 39

[33] R. Giering and T. Kaminski, "Recipes for adjoint code construction," *ACM Transactions on Mathematical Software*, vol. 24, no. 4, pp. 437–474, 1998. 31

[34] L. Nardi, C. Sorror, F. Badran, and S. Thiria, "Yao: a software for variational data assimilation using numerical models," in *Computational Science and Its Applications*, pp. 621–636. Springer, 2009. 31

[35] L. Hascoët and V. Pascual, "Tapenade 2.1 user's guide," Technical Report 0300, INRIA, 2004. 31

[36] D. Béréziat and I. Herlin, "Solving ill-posed image processing problems using data assimilation," Research Report RR-6879, Inria, March 2009. 32

[37] D.S. Oliver, "Calculation of the inverse of the covariance," *Mathematical Geology*, vol. 30, no. 7, pp. 911–933, 1998. 34

[38] J. L. Morales and J. Nocedal, "Remark on algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization," *ACM Transaction on Mathematical Software*, vol. 38, no. 1, pp. 7:1–7:4, December 2011. 39

[39] P. Courtier, J.-N. Thépaut, and A. Hollingsworth, "A strategy for operational implementation of 4D-Var, using and incremental approach," *Quaterly Journal of the Royal Meteorological Society*, vol. 120, no. 1367–1387, 1994. 42

[40] E. Valur Hólm, "Lectures notes on assimilation algorithms," Tech. Rep., European Centre for Medium-Range Weather Forecasts Reading, U.K, April 2008. 42

[41] D. Béréziat and I. Herlin, "Image-based modelling of ocean surface circulation from satellite acquisitions," in *International Conference on Computer Vision Theory and Applications*, January 2014. 42, 136, 141

[42] I. Herlin, D. Béréziat, and N. Mercier, "Improvement of motion estimation by assessing the errors on the evolution equation," in *International Conference on Computer Vision Theory and Applications*, Rome, Italy, February 2012, vol. 2, pp. 235–240. 42

[43] 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. 45

[44] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986. 45

[45] J.A. Sethian, *Level Set Methods*, Cambridge University Press, 1996. 48

[46] 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. 48

[47] M. Sussman, E. Fatemi, P. Smereka, and S. Osher, "An improved level set method for incompressible two-phase flows," *Computers & Fluids*, vol. 27, no. 5 6, pp. 663 – 680, 1998. 48

[48] 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. 48, 49

[49] V. Chabot, M. Nodet, N. Papadakis, and A. Vidard, "Accounting for observation errors in image data assimilation," *Tellus Series A : Dynamic meteorology and oceanography*, vol. 67, no. 23629, pp. 19, February 2015. 52, 142

[50] A. T. Weaver and S. Ricci, "Constructing a background-error correlation model using generalized diffusion operators," *ECWMF procreedings on "Recent developments in data assimilation for atmosphere and ocean*, 2003. 55

[51] J. Parent-du Châtelet, "Aramis, le réseau franccais de radars pour la surveillance des précipitations," *Société météorologique de France*, 2003. 79

[52] S. Mecklenburg, A. Jurczyk, J. Szturc, and K Osrodka, "Quantitative precipitation forecasts (QPF) based on radar data for hydrological models," in *COST action*, 2002, vol. 717. 80

[53] E. Ebert, L. Wilson, B. Brown, P. Nurmi, H. Brooks, J. Bally, and M. Jaeneke, "Verification of nowcasts from the WWRP Sydney 2000 forecast demonstration project," *Weather and Forecasting*, vol. 19, no. 1, pp. 73–96, 2004. 80

[54] A. Neumann, *Introduction d'outils de l'intelligence artificielle dans la prévision de pluie par radar*, Thesis, Ecole Nationale des Ponts et Chaussées, December 1991. 81

[55] M. Dixon and Wiener G., "TITAN: Thunderstorm Identification, Tracking, Analysis, and Nowcasting—A Radar-based Methodology," *Journal of Atmospheric and Oceanic Technology*, vol. 10, no. 6, pp. 785–797, 1993. 81

[56] D. Sempere-Torres, R. Sánchez-Diezma, M. Córdoba, R. Pascual, and I. Zawadzki, "An operational methodology to control radar measurements stability from mountain returns," *Conference on Radar Meteorology of the American Meteorological Society*, vol. 30, pp. 264, 2001. 83

[57] J. S. Marshall and W. M. K. Palmer, "The distribution of raindrops with size," *Journal of meteorology*, vol. 5, pp. 165–166, 1948. 83

[58] J. Van Baelen, F. Tridon, and Y. Pointin, "Simultaneous X-band and K-band study of precipitation to derive specific Z-R relationships," *Atmospheric Research*, vol. 94, no. 4, pp. 596–605, 2009. 84

[59] C. Temperton, M. Hortal, and A. Simmons, "A two-time-level semi-Lagrangian global spectral model," *Quarterly Journal of the Royal Meteorological Society*, vol. 127, no. 571, pp. 111–127, 2001. 88

[60] M. Hortal, "The development and testing of a new two-time-level semi-Lagrangian scheme (SETTLS) in the ECMWF forecast model," *Quarterly Journal of the Royal Meteorological Society*, vol. 128, no. 583, pp. 1671–1687, 2002. 88

[61] P. Sakov and P. R. Oke, "A deterministic formulation of the ensemble Kalman filter: An alternative to ensemble square root filters," *Tellus Series A : Dynamic meteorology and oceanography*, 2008. 98, 110

[62] P. L. Houtekamer and H. L. Mitchell, "Data assimilation using an ensemble Kalman filter technique," *Monthly Weather Review*, vol. 128, pp. 796–811, 1998. 98

[63] J. L. Anderson and S. L. Anderson, "A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts," *Monthly Weather Review*, vol. 127, pp. 2741—2758, 1999. 98

[64] T. M. Hamill, Jeffrey S. Whitaker, and C. Snyder, "Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter," *Monthly Weather Review*, vol. 129, pp. 2776–2790, 2001. 98, 113, 144

[65] P. R. Oke, P. Sakov, and S. P. Corney, "Impacts of localisation in the EnKF and EnOI: experiments with a small model," *Ocean Dynamics*, vol. 57, no. 1, pp. 32–45, 2007. 98

[66] J. L. Anderson, "An adaptive covariance inflation error correction algorithm for ensemble filters," *Tellus Series A : Dynamic meteorology and oceanography*, vol. 59, pp. 210—-224, 2007. 98

[67] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *Journal of Geophysical Research*, 1994. 104

[68] G. Evensen, *Data Assimilation -The Ensemble Kalman Filter*, Springer, 2006. 105, 112, 115

[69] O. Mac Aodha, G.J. Brostow, and M. Pollefeys, "Segmenting video into classes of algorithm-suitability," in *Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1054–1061. 107

[70] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision*, Springer-Verlag, Ed., Prague, Czech Republic, May 2004, vol. 4, pp. 25–36. 107

[71] M. J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75 – 104, 1996. 107

[72] Y. Li and S. Osher, "A new median formula with applications to PDE based denoising," *Communication in Mathematical Sciences*, vol. 7, no. 3, pp. 741–753, 2009. 108

[73] E. Agullo, J. Demmel, J. Dongarra, B. Hadri, J. Kurzak, J. Langou, H. Ltaief, P. Luszczek, and S. Tomov, "Numerical linear algebra on emerging architectures: The PLASMA and MAGMA projects," *Journal of Physics: Conference Series*, vol. 180, no. 1, pp. 012037, 2009. 115

[74] P. L. Houtekamer and H. L. Mitchell, "A sequential ensemble Kalman filter for atmospheric data assimilation," *Monthly Weather Review*, vol. 129, pp. 123–137, 1998. 117

[75] B. R Hunt, E. J. Kostelich, and I. Szunyogh, "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter," *Physica D: Nonlinear Phenomena*, vol. 230, no. 1, pp. 112–126, 2007. 117, 124

[76] L. Nerger, S. Danilov, W. Hiller, and J. Schröter, "Using sea-level data to constrain a finite-element primitive-equation ocean model with a local SEIK filter," *Ocean Dynamics*, vol. 56, no. 5-6, pp. 634–649, 2006. 117

[77] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a tree traversal algorithm," *Journal ACM*, vol. 23, no. 2, pp. 368–388, April 1976. 125

[78] R. Ohlander, K. Price, and D. R. Reddy, "Picture segmentation using a recursive region splitting method," *Computer Graphics and Image Processing*, vol. 8, no. 3, pp. 313–333, 1978. 125

[79] T. Janjić, L. Nerger, A. Albertella, J. Schröter, and S. Skachko, "On domain localization in ensemble-based Kalman filter algorithms," *Monthly Weather Review*, vol. 139, no. 7, pp. 2046–2060, 2011. 127, 144

[80] http://i21www.ira.uka.de/image_sequences/, "Nibelungen-platz," 1995. 135

[81] C. H. Bishop, Etherton B. J., and Majumdar S. J., "Adaptive sampling with the ensemble transform Kalman filter. part i: Theoretical aspects," *Monthly Weather Review*, vol. 129, no. 3, pp. 420–436, 2001. 144

[82] T. Miyoshi, S. Yamane, and T. Enomoto, "Localizing the error covariance by physical distances within a local ensemble transform Kalman filter (LETKF)," *Scientific Online Letters on the Atmosphere*, vol. 3, pp. 89–92, 2007. 144