HAL
archives-ouvertes.fr

An Image-Based Ensemble Kalman Filter for Motion Estimation

Yann Lepoittevin, Isabelle Herlin, Dominique Béréziat

An Image-Based Ensemble Kalman Filter for Motion Estimation

Yann Lepoittevin[1] , Isabelle Herlin[1] and Dominique Béréziat[2,3]

[1] *Inria Paris-Rocquencourt, Domaine de voluceau, 78150, Le Chesnay, France*

[2]*LIP6, Sorbonne Universities, UPMC Univ Paris 06, UMR 7606, 75005, Paris, France*

[2]*LIP6, CNRS, UMR 7606, Paris, 75005, France*

Abstract: This paper designs an Image-based Ensemble Kalman Filter (IEnKF), whose components are defined only from image properties, to estimate motion on image sequences. The key elements of this filter are, first, the construction of the initial ensemble, and second, the propagation in time of this ensemble on the studied temporal interval. Both are analyzed in the paper and their impact on results is discussed with synthetic and real data experiments. The initial ensemble is obtained by adding a Gaussian vector field to an estimate of motion on the first two frames. The standard deviation of this normal law is computed from motion results given by a set of optical flow methods of the literature. It describes the uncertainty on the motion value at initial date. The propagation in time of the ensemble members relies on the following evolution laws: transport by velocity of the image brightness function and Euler equations for the motion function. Shrinking of the ensemble is avoided thanks to a localization method and the use of observation ensembles, both techniques being defined from image characteristics. This Image-based Ensemble Kalman Filter is quantified on synthetic experiments and applied on traffic and meteorological images.

1 INTRODUCTION

This paper revisits the use of optical flow methods to estimate motion on discrete image sequences. Based on the concept of data assimilation the filter relies on 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 for instance in (Papadakis et al., 2007; Titaud et al., 2010; Béréziat and Herlin, 2011). However, these approaches present some major restrictions:

- 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 computation time than a single integration of the model, as done with a filtering approach.

- they provide an estimate of motion fields on the whole time interval, from the image sequence, but no uncertainty measure is at hand. That uncertainty value is however mandatory for further in-

terpretation of motion results.

The design of filtering methods, and in particular the ones based on the Ensemble Kalman Filter (EnKF) (Evensen, 2003), is an alternative that is not affected by the previous limitations. EnKF is an adaptation of the Kalman filter (Kalman, 1960) that does not use an analytical description of the Probability Density Function (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 PDF is described by the spread of the ensemble.

As experiments are done with a small-size ensemble (an ensemble of 40 members is constructed for the experiments described in the paper), in order to keep the computational cost as low as possible, the values of the different members usually come closer one to each other during temporal integration: this is named "shrinking of the ensemble". Several approaches are used to avoid this deficiency. Inflation and localization methods (Anderson and Anderson, 1999; Hamill et al., 2001; Oke et al., 2007) either increase the variance values or decrease the covariance values of the matrices computed on the ensemble members before the analysis step. Other methods, such as the deterministic EnKF (Sakov and Oke, 2008), correct the

members obtained after the analysis step so that the covariance matrices computed from these members become closer to the ones modeled with the Kalman filter. Beyou et al. (Beyou et al., 2012) apply EnKF for motion estimation. Each motion member is written as the sum of a mean and a perturbation. When computing the analysis, this formalism involves an additional inversion of the observation error covariance matrix. Shrinking is avoided by resampling the ensemble before further integration. In our paper, the methods chosen to solve the shrinking issue are only based on image properties and should not lead to prohibitive computational time. 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 (Anderson, 2007). However, this inflation parameter may not be directly defined from image properties. The deterministic EnKF computes the covariance matrix associated to the perturbations after the analysis step, conducting to an overload of computations. Due to the resampling process applied in (Beyou et al., 2012), the error covariance matrices, computed from the ensemble members, do not characterize the uncertainty on motion estimation. Consequently, these three approaches are not used in the following. In order to avoid shrinking, observation ensembles (Houtekamer and Mitchell, 1998) are first constructed, based on characteristics of image acquisitions. This is described in Subsection 3.4. Second, an innovative localization process is defined to limit spurious covariances between pixels. It is based on: 1- distance between pixels (points separated by a high distance are usually independent), 2- similarities between pixels (points belonging to the same objects or structures are usually highly correlated). It is described in Subsection 3.3.

Section 2 defines the mathematical notations and summarizes the EnKF algorithm, while Section 3 describes the characteristics of the method discussed in the paper: an original method for constructing the ensemble, evolution laws and propagation in time of the ensemble. Section 4 discusses the results obtained on a synthetic experiment and on real data.

2 ENSEMBLE KALMAN FILTER

As said before, the aim of this research work is to produce a dense-in-time motion estimation from a sparse-in-time discrete image sequence.

2.1 Mathematical Setting

Let define the following notations that are used in the remaining of the paper. $\mathbf{x} = \begin{pmatrix} x & y \end{pmatrix}^T$ is a pixel belonging to the image domain Ω, with $.^T$ the transpose operator. The sequence of N^O images $\{ I_l^O, \quad l \in 1..N^O \}$ is defined on Ω. It has been acquired on a given temporal interval $0..T$.

The data assimilation method requires the definition of a state vector $\mathbf{X}$ whose value at time index k is $\mathbf{X}_k = \begin{pmatrix} \mathbf{w}_k^T & I_k \end{pmatrix}^T$. $\mathbf{w} = \begin{pmatrix} u & v \end{pmatrix}^T$ denotes the motion field and I is a synthetic image field with the same physical properties than the studied image sequence. If an image acquisition is available at index k, it is used to compute the observation vector $\mathbf{Y}_k$. The aim of data assimilation is to get an estimate, or analysis, $\mathbf{X}_k^{(a)}$ of the true state vector $\mathbf{X}_k^{(r)}$ from the so-called background state vector $\mathbf{X}_k^{(b)}$ and, if available, the observation vector $\mathbf{Y}_k$.

Ensemble methods rely on a number of members that evolve simultaneously in time. $\mathbf{X}_k^j$ denotes the state vector at index k of the jth member of the ensemble. If an observation ensemble is constructed from images, $\mathbf{Y}_k^j$ is the jth member of this observation ensemble at index k.

$\overline{.}$ denotes the mean over the ensemble members.

2.2 Algorithm

The Kalman Filter (Kalman, 1960) performs a time integration of the model: it provides an approximation of the true state and an uncertainty value of this estimation, given the following elements:

- A background value $\mathbf{X}_0^{(b)}$. Its uncertainty is described by a PDF, which is supposed to be a zero-mean normal law with covariance matrix $\mathbf{B}_0^{(b)}$.

- Observations $\mathbf{Y}_k$, at some time indexes k for which images are available. Their uncertainty is described by zero-mean normal laws of covariance matrices R_k.

- At index k, the state vector $\mathbf{X}_k$ and the observation vector $\mathbf{Y}_k$ are compared thanks to an observation operator $I\!H$, such that $I\!H(\mathbf{X})$ belongs to the observation space.

- Time integration of the state vector is obtained from a linear model $I\!M$:

$$\mathbf{X}_k = I\!M \big(\mathbf{X}_{k-1} \big) . \tag{1}$$

Equations of the Kalman filter, in case of a linear observation operator $I\!H$, are the following:

1. At time index k, the background value $\mathbf{X}_k^{(b)}$ is obtained from the estimation (also named analysis) $\mathbf{X}_{k-1}^{(a)}$ at index $k-1$ by applying:

$$\mathbf{X}_k^{(b)} = I\!\!M\big(\mathbf{X}_{k-1}^{(a)}\big) \; . \qquad (2)$$

The propagation in time of the uncertainty covariance matrix $\mathbf{B}_k^{(b)}$ satisfies:

$$\mathbf{B}_k^{(b)} = I\!\!M\mathbf{B}_{k-1}^{(a)}I\!\!M^T \; . \qquad (3)$$

For sake of simplicity, $I\!\!M$ denotes both the linear model and its associated matrix.

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

3. If an observation vector $\mathbf{Y}_k$ is available at k, then the analysis is given by the analysis equation:

$$\mathbf{X}_k^{(a)} = \mathbf{X}_k^{(b)} + \mathbf{K}\big(\mathbf{Y}_k - I\!\!H\mathbf{X}_k^{(b)}\big) \; , \qquad (4)$$

where :

$$\mathbf{K} = \mathbf{B}_k^{(b)}I\!\!H^T\big(I\!\!H\mathbf{B}_k^{(b)}I\!\!H^T + R_k\big)^{-1} \; , \qquad (5)$$

with $I\!\!H$ denoting both the linear operator and its associated matrix. The uncertainty covariance matrix verifies:

$$\mathbf{B}_k^{(a)} = \mathbf{B}_k^{(b)} - \mathbf{K}I\!\!H\mathbf{B}_k^{(b)} \; . \qquad (6)$$

Having summarized the Kalman filter equations, two major issues have to be discussed. First, time propagation of the background covariance matrix $\mathbf{B}_k^{(b)}$ with Eq. (3), by the linear model $I\!\!M$, leads to prohibitive computational requirements for large-sized state vectors. Second, if $I\!\!M$ is non linear, its approximation by its tangent linear model M is used for the propagation of the uncertainty in Eq. (3). Consequently, the resulting value $\mathbf{B}_k^{(b)}$ is only an approximation of the true value. In this paper, the model $I\!\!M$, expressing the Lagrangian constancy of velocity, includes non linear equations and its propagation is affected by the approximation. These two issues of computational burden and approximation are avoided by the Ensemble Kalman filter (Evensen, 2003). An ensemble of background state vectors $\mathbf{X}_k^{(b),j}$ is defined at each time index k. It samples the uncertainty covariance matrix $\mathbf{B}_k^{(b)}$. Let denote $\mathbf{X}_k^{(b)} = \overline{\mathbf{X}_k^{(b),j}}$ the mean over the ensemble members. An approximation of the covariance matrix is obtained from the ensemble using the equation:

$$\mathbf{B}_k^{(b)} \approx \overline{(\mathbf{X}_k^{(b),j} - \mathbf{X}_k^{(b)})(\mathbf{X}_k^{(b),j} - \mathbf{X}_k^{(b)})^T} \; . \qquad (7)$$

Initialized at initial date 0, the ensemble is propagated in time by integrating each member using the model $I\!\!M$ as in Eq. (1). If no observation is available at time index k, the estimation $\mathbf{X}_k^{(a)}$ is set equal to $\mathbf{X}_k^{(b)}$ and the uncertainty is approximated using Eq. (7). If an observation is available at k, an analysis is computed for each member j by:

$$\mathbf{X}_k^{(a),j} = \mathbf{X}_k^{(b),j} + \mathbf{K}(\mathbf{Y}_k - I\!\!H\mathbf{X}_k^{(b),j}) \; , \qquad (8)$$

where :

$$\mathbf{K} = \mathbf{B}_k^{(b)}I\!\!H^T\big(I\!\!H\mathbf{B}_k^{(b)}I\!\!H^T + R_k\big)^{-1} \; . \qquad (9)$$

The estimation is defined as $\mathbf{X}_k^{(a)} = \overline{\mathbf{X}^{(a),j}}$ and its uncertainty is approximated by replacing $^{(b)}$ by $^{(a)}$ in Eq. (7). It should be noted that all members are involved in the computation of $\mathbf{B}_k^{(b)}$ and consequently in the estimation of each element $\mathbf{X}_k^{(a),j}$.

The ensemble Kalman filter is naturally parallelizable, as time integration of each member is independent from the others. Moreover, matrix products and inversions are done by the parallel linear algebra library PLASMA (Agullo et al., 2009) that enables to significantly reduce computational costs.

3 IMPLEMENTATION OF AN ENSEMBLE KALMAN FILTER FOR MOTION ESTIMATION

This section describes the core of this paper: the design of an ensemble Kalman filter for motion estimation from an image sequence. It is named Image-based Ensemble Kalman Filter or IEnKF.

3.1 Evolution Model

As previously explained, the ensemble members are integrated in time by the numerical model $I\!\!M$ obtained by time discretization of a continuous evolution model $I\!\!M_c$. The heuristics for the evolution of motion and images are the following:

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

$$\frac{d\mathbf{w}}{dt} = \frac{\partial \mathbf{w}}{\partial t} + (\mathbf{w} \cdot \nabla)\mathbf{w} = 0 \; . \qquad (10)$$

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

$$\frac{\partial I}{\partial t} + \mathbf{w} \cdot \nabla I = 0 \; . \qquad (11)$$

The state vector is defined by $\mathbf{X} = \begin{pmatrix} \mathbf{w}^T & I \end{pmatrix}^T$. Eqs. (10) and (11) are then summarized by:

$$\frac{\partial \mathbf{X}}{\partial t} + I\!M_c(\mathbf{X}(t)) = 0 \ . \qquad (12)$$

Temporal discretization of Eq. (12) with an explicit Euler scheme leads to:

$$\mathbf{X}_k = I\!M(\mathbf{X}_{k-1}) \qquad (13)$$

that is applied for propagating each member of the ensemble.

3.2 Ensemble Construction

The motion estimation method highly depends on the design of the initial ensemble at the beginning of the image sequence. This ensemble should span a vectorial space that contains the truth and properly samples the uncertainties. It is usually obtained by adding some perturbations to an initial guess with a Monte Carlo method. This subsection describes an innovative alternative where the ensemble is designed from the image observations.

The state of each ensemble member is composed of a motion field and an image. However, there is no need to construct an ensemble of images, as the image component is only used as a display of the underlain dynamics. Consequently, at initial date, the members include the same image, the first acquisition, but various motion fields.

There are lots of methods available in order to approximate motion between two observations, see for instance (Sun et al., 2010; Baker et al., 2011) for a survey on the subject. The approaches differ in their formulation of the optical flow, the spatial smoothness assumptions, the implementation tools (coarse-to-fine estimation, texture decomposition, median filtering, etc.). For each approach, various results are obtained according to 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 the different regions of the image domain. This 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 as in (Mac Aodha et al., 2010). Being in the same spirit, we came to the conclusion that generating motion fields from a large set of codes, varying formulation and parametrization, is the best way to ensure spanning a vector space of motion fields including the true motion field and to assess the uncertainty on estimation.

The optical flow algorithms, applied in the paper, are variational methods that rely on the brightness constancy hypothesis (Horn and Schunk, 1981). They estimate motion by minimizing a cost function, which includes, at least, a data term and a weighted regularization term. Modifying this weight value allows the user to choose smoothness properties of the solution. A small value conducts to a result strongly relying on the data and containing outliers linked to acquisition noise, whereas a higher one produces smooth solutions. Three types of norm are implemented to compute the cost function: the quadratic norm (Horn and Schunk, 1981), the Charbonnier norm (Brox et al., 2004) and the Lorentzian norm (Black and Anandan, 1996). For each method, a coarse-to-fine approach is used in order to converge while minimizing the cost function. Before each resolution change, the motion field at the current scale is smoothed by a median filter, whose size is also a parameter. Its value impacts the quality of the solution as discussed in (Sun et al., 2010). Another possibility to control smoothness of the motion field is to integrate this median filtering directly in the cost function. A weight is then given to the corresponding additional term as in (Li and Osher, 2009).

Motion is estimated between the first two images with this set of methods and various parametrizations concerning the regularization weight, the norm type, the filter type (median or weighted median). Such motion result is denoted $\mathbf{m}^j$ with j spanning the set of methods. Let denote $\mathbf{m} = \overline{\mathbf{m}^j}$ the mean of results. An uncertainty measure on motion is defined by the covariance matrix:

$$\mathbf{B} = \overline{(\mathbf{m}^j - \mathbf{m})(\mathbf{m}^j - \mathbf{m})^T} \ . \qquad (14)$$

The motion field corresponding to the mean of the ensemble is then chosen as initial value. The motion ensemble is constructed from this value and the normal law associated to the covariance matrix $\mathbf{B}$. One member is denoted $\mathbf{w}^j$. The size of that ensemble is 40 in the experiments.

3.3 Ensemble Propagation

As stated in Section 1, a localization method is used in order to limit the shrinking that occurs after computing analysis with Eqs. (8) and (9). This localization corrects the spurious covariances that appear when computing the covariance matrix from a small-sized ensemble with Eq. (7). These covariances link pixels that should be independent because too far one from each other or belonging to different objects of the image. Let ρ denote the localization matrix and $\circ$ the point-wise matrix product (also known as Hadamard product). The localization process is applied before computing the analysis from background and obser-

vations. Let define

$$\mathbf{L}_k^{(b)} = \rho \circ \mathbf{B}_k^{(b)} \qquad (15)$$

and replace $\mathbf{B}_k^{(b)}$ by $\mathbf{L}_k^{(b)}$ in the analysis equations Eqs. (8) and (9). We obtain the following equations:

$$\mathbf{X}_k^{(a),j} = \mathbf{X}_k^{(b),j} + \mathbf{K}_L(\mathbf{Y}(k) - I\!\!H\mathbf{X}_k^{(b)j}) \quad (16)$$

where :

$$\mathbf{K}_L = \mathbf{L}_k^{(b)} I\!\!H^T \left(I\!\!H\mathbf{L}_k^{(b)} I\!\!H^T + R_k \right)^{-1}, \qquad (17)$$

that compute the analysis $\mathbf{X}_k^{(a),j}$ of member j at time index k.

The main group of methods for defining the ρ matrix, denoted ρ_d (where the subscript d stands for distance), is defined according to distances between pixels, as in (Hamill et al., 2001; Haugen and Evensen, 2002; Brankart et al., 2003). ρ_d is thus designed so that pixels separated by a distance higher than a given threshold get an almost null correlation. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ denote two pixels of the image domain, ρ_d depends on their distance $d_{12} = ||\mathbf{x}_1 - \mathbf{x}_2||_2$ and is defined by:

$$\rho_d(\mathbf{x}_1, \mathbf{x}_2) = \left(1 + \frac{d_{12}}{a}\right) \times \exp\left(-\frac{d_{12}}{a}\right). \qquad (18)$$

The parameter a, involved in Eq. (18), is called decorrelation distance. The values of ρ_d, depending on d_{12}, are displayed on Fig. 1 for three values of a.
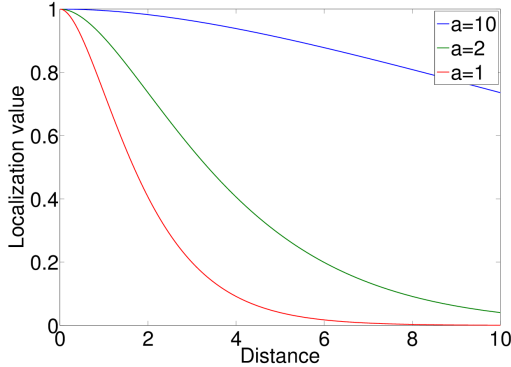

Figure 1: Values of ρ_d, for $a = 1, 2, 10$, as a function of the distance between pixels.

We propose a method that also nullifies correlation between pixels that belong to different objects. The localization matrix ρ_s (where the subscript s stands for similarity) depends on pixels similarities. Let denote I_1 and I_2 the gray values of pixels $\mathbf{x}_1$ and $\mathbf{x}_2$ and $s_{12} = ||I_1 - I_2||_2$. ρ_s is defined by:

$$\rho_s(l_1, l_2) = \left(1 + \frac{s_{12}}{a_l}\right) \times \exp\left(-\frac{s_{12}}{a_l}\right). \qquad (19)$$

Our innovative localization matrix ρ, used in Eq. (15), is then defined as a function of ρ_d and ρ_s. Consequently, the correlation values between pixels that are far apart or belong to different objects are almost null.

3.4 Observations

As defined in Subsection 2.1, the state vector is written $\mathbf{X} = \left(\mathbf{w}^T \quad I\right)^T$. The synthetic image I satisfies the assumption that brightness values are transported by velocity, as explained in Subsection 3.1. The observation operator $I\!\!H$ is defined as the linear projection on the second component of the state vector:

$$I\!\!H(\mathbf{X}) = I. \qquad (20)$$

This observation operator is used during the assimilation process to compare the function I with the image acquisitions: the motion field is estimated in order minimize the discrepancy between them.

As explained in Section 1, observation ensembles are used in order to avoid shrinking when propagating the ensemble members in time. In the case of motion estimation from images, the formulation of the observation ensemble associated to one image acquisition is straightforward. Information on the sensor characteristics allows the user to model the acquisition noise as an additive zero-mean Gaussian noise, whose standard deviation is usually included in the metadata associated to the sensor. This is written as:

$$I_k^O = I_k^{(r)} + \mathcal{N}(0, \sigma), \qquad (21)$$

where I_k^O is the acquired image of the unknown truth $I_k^{(r)}$. The image ensemble associated to I_k^O is then obtained with the same heuristics. Each member I_k^j is obtained by adding a gaussian noise to the acquired image:

$$I_k^j = I_k^O + \mathcal{N}(0, \sigma). \qquad (22)$$

Experiments showed that an ensemble of 20 elements is sufficient to correctly sample the error law and avoid shrinking. Considering additional members increases the memory requirements without decisive improvement on results. Having less members decreases the accuracy of motion estimation.

4 RESULTS ON MOTION ESTIMATION

4.1 Synthetic Experiment

Starting from the initial image and motion field, displayed on Fig. 2, the model $I\!\!M$ is integrated in time, as explained in Subsection 3.1. Motion fields obtained from that integration are used as ground-truth for evaluating results. Six image snapshots are taken every ten time steps and displayed on Fig. 3. They
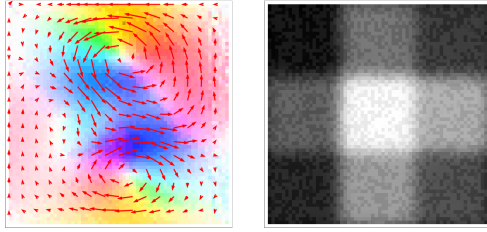
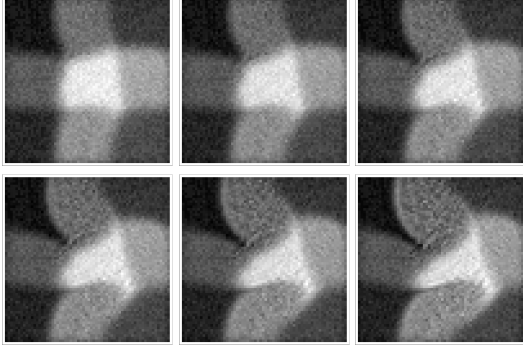Figure 2: Initial conditions of ground-truth.

Figure 3: Snapshots of the synthetic experiment.

are the image observations of the assimilation experiment. The observation ensemble is constructed from these data as explained in Subsection 3.4. A motion ensemble of 40 members is created as explained in Subsection 3.2.

IEnKF is then applied to estimate motion either without localization or with localization during computation of the analysis using respectively Eq. (8) or Eq. (16). The decorrelation distance that is used for the localization process has a value of 1.

Errors statistics on the discrepancy between estimation and ground-truth are computed and analyzed. The one concerning the angular error is obtained by taking the spatial Average of the Absolute values of Angular Errors (AAAE). First, AAAE are computed for 20 members (randomly chosen among the set of 40) and visualized as blue points, at each date of the studied temporal interval, on Fig. 4. Second, AAAE is computed for the ensemble mean and represented as a red curve on Fig. 4. Figures on norm errors are similar. As it can be seen, the initial ensemble is widely spread and the first analysis step, at date 10, significantly reduces the errors. One can see the effects of localization on the bottom figure, compared to the top one. First, the ensemble is spread enough for a longer time. When the shrinking is important, the error covariance matrix becomes small. At the limit, when the matrix is almost null, the analysis step has no more effect on the ensemble (as it can be seen in the analysis formula, Eq. (8)). Keeping an ensemble spread large
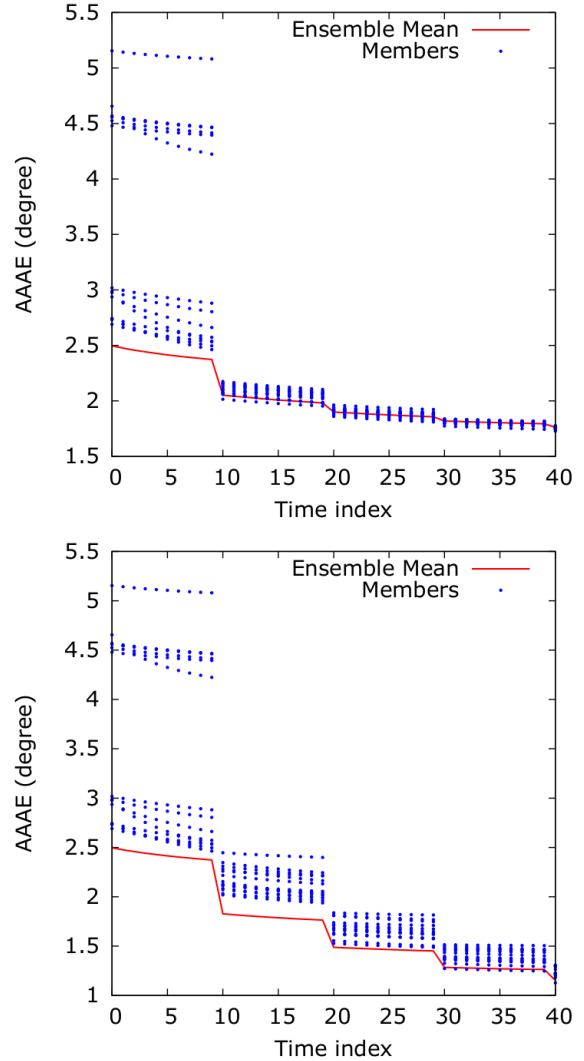

Figure 4: spatial Average of the Absolute values of Angular Errors (AAAE) over time without localization (*top*) and with localization (*bottom*). The blue points correspond to the AAAE of members of the ensemble and the red curve to the AAAE of the ensemble mean.

enough, as long as possible, is then of major interest for online image processing. Second, 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, the accuracy of the estimation still improves at the 5th or 6th analysis step with localization, whereas no effect occurs anymore after the 4th computation of an analysis without localization.

Fig. 5 displays, on top, the norm of the error between the estimation and the truth and, on the bottom, the variance of the ensemble results. It can be

observed that the spread of the ensemble has a good geographical correspondance with the errors.
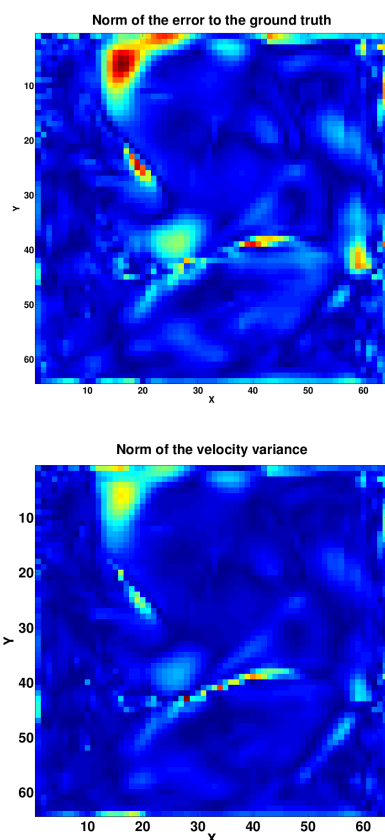

Figure 5: *Top*: Norm of the error between the estimation and the ground truth. *Bottom*: Variance of the ensemble results.

If computing motion between the 5th and 6th observations with the whole set of motion estimation algorithms (see Subsection 3.2) and analyze the results, the best method shows an average angular error of 5 degrees, to be compared with the 1-degree error obtained by our method.

4.2 Experiment on Real Data

IEnKF was applied on traffic images from the database KOGS/IAKS of the Karlsruhe University (http://i21www.ira.uka.de/image_sequences/, 1995). The previously described implementation of IEnKF is used in order to estimate motion of cars. As quality assessment is not at hand with arrow or color displays, results are visualized on Fig. 6, thanks to trajectories of characteristic points. Each trajectory is obtained by advecting the chosen point with the estimated motion field. On each image of the figure, a colored circle displays the current position of one

Figure 6: Motion results given as trajectories, with the position (*circle*) of a characteristic point on the first frame (*top*) or the last one (*bottom*).
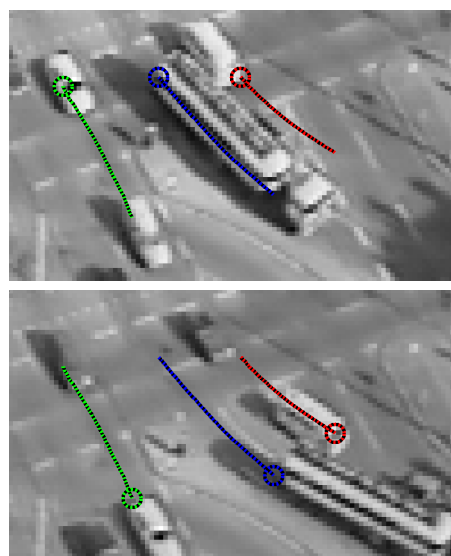

Figure 7: Motion results given as trajectories, with the position (*circle*) of a characteristic point on the first frame (*top*) or the last one (*bottom*).

characteristic point and the corresponding curve its whole trajectory. As it can be seen, motion is correctly estimated, in norm and orientation, for the taxi and the van coming from the right. This conclusion is still valid when defining new characteristic points. The black car on the left gets weaker quality motion results due to its low contrast.

A second sequence of the same data base has been processed by IEnKF and results are given on Fig. 7 with trajectories of characteristic points displayed on the first and last images. As cars are accelerating when the traffic light turns green, the algorithm slightly underestimates their speed. A future
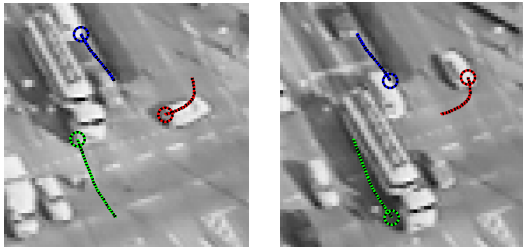
Figure 8: Motion results given as trajectories, with the position (*circle)* of a characteristic point on the first frame (*left)* or the last one (*right)*.
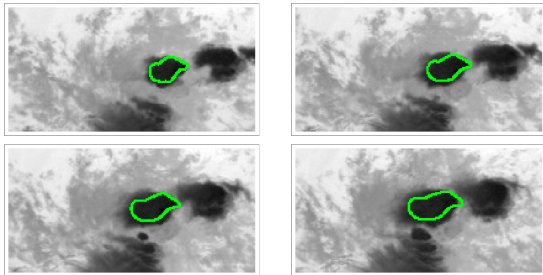

Figure 9: Advection of the boundary by the estimated motion fields.

implementation of the method will involve an acceleration term in the state vector in order to improve the estimation.

Results obtained on a third sequence of this database are displayed on Fig. 8. The computed trajectories of three characteristic points are shown and one can visualize that one car undertakes a U-turn.

IEnKF was also applied on meteorological satellite data where images are acquired every 15 minutes in the infrared domain with a 5 km resolution. The foreseen operational application is to improve short term forecast of cloud cover. A first experiment concerns a sequence displaying convective cells. Results are given on Fig. 9 by advecting the boundary of one cloud with the estimated motion field. One can see that these points remain on the boundary of the convective cell during time integration. This accurate tracking confirms that the cell velocity is well estimated.

IEnKF was applied on a second meteorological sequence. Results are given on Fig. 10 by advecting the boundary of one cloud with the estimated motion field. This again demonstrates correctness of the estimation.
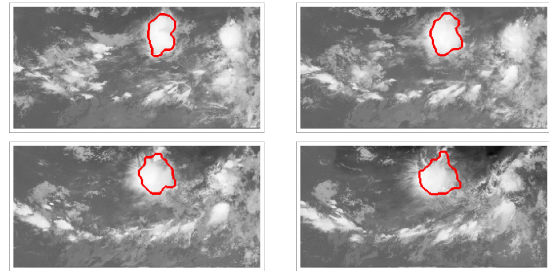

Figure 10: Advection of the boundary by the estimated motion fields.

5 CONCLUSION

The paper describes a motion estimation method, based on the ensemble Kalman filter, which is named IEnKF. The initial motion ensemble is defined from image properties and a number of optical flow methods from the literature. The usual shrinking of the ensemble is delayed thanks to the construction of observation ensembles and to a localization method, both defined from image properties. The approach was tested on synthetic experiments whose ground-truth allows quantifying results. It was also applied on two major applications. On traffic images, the approach allows a correct estimation of motion, which is demonstrated by computing trajectories of characteristic points on moving cars. On meteorological satellite acquisitions, the method demonstrates its performance with the correct tracking of clouds boundary.

Future work concerns the extension of the state vector to take into account acceleration and a mathematical description of objects and structures displayed on the sequence. Both components will impact the estimation of motion. For being able to process images online, two main issues will be investigated as, for now, computation on a 100×100 pixels image takes approximately 10 minutes and several gigabytes of active memory. First, on the methodological part, model reduction is foreseen with small-size motion and image subspaces. The reduced state vector will then contains around 10-20 components and application of IEnKF will be at low memory cost. Second, all components of code will be fully parallelized: IEnKF, computation of motion and image bases, and model reduction.

ACKNOWLEDGEMENTS

REFERENCES

Agullo, E., Demmel, J., Dongarra, J., Hadri, B., Kurzak, J., Langou, J., Ltaief, H., Luszczek, P., and Tomov, S. (2009). Numerical linear algebra on emerging architectures: The PLASMA and MAGMA projects. *Journal of Physics: Conference Series*, 180(1):012037.

Anderson, J. L. (2007). An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus A*, 59:210–224.

Anderson, J. L. and Anderson, S. L. (1999). A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127:2741–2758.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R. (2011). A Database and Evaluation Methodology for Optical Flow. *International Journal on Computer Vision*, 92(1):131.

Béréziat, D. and Herlin, I. (2011). Solving ill-posed image processing problems using data assimilation. *Numerical Algorithms*, 56(2):219–252.

Beyou, S., Cuzol, A., Gorthi, S. S., and Mémin, E. (2012). Weighted ensemble transform Kalman filter for image assimilation. *Tellus A*, 65.

Black, M. J. and Anandan, P. (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75 – 104.

Brankart, J.-M., Testut, C.-E., Brasseur, P., and Verron, J. (2003). Implementation of a multivariate data assimilation scheme for isopycnic coordinate ocean models: Application to a 1993–1996 hindcast of the North Atlantic Ocean circulation. *Journal of Geophysical Research: Oceans*, 108.

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

Evensen, G. (2003). The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367.

Hamill, T. M., Whitaker, J. S., and Snyder, C. (2001). Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review*, 129:2776–2790.

Haugen, V. E. J. and Evensen, G. (2002). Assimilation of SLA and SST data into an OGCM for the Indian Ocean. *Ocean Dynamics*, 52(3):133–151.

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

Houtekamer, P. L. and Mitchell, H. L. (1998). Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 128:796–811.

http://i21www.ira.uka.de/image_sequences/ (1995). Nibelungen-platz.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45.

Li, Y. and Osher, S. (2009). A new median formula with applications to PDE based denoising. *Communication in Mathematical Sciences*, 7(3):741–753.

Mac Aodha, O., Brostow, G., and Pollefeys, M. (2010). Segmenting video into classes of algorithm-suitability. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1054–1061.

Oke, P. R., Sakov, P., and Corney, S. P. (2007). Impacts of localisation in the EnKF and EnOI: experiments with a small model. *Ocean Dynamics*, 57(1):32–45.

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

Sakov, P. and Oke, P. R. (2008). A deterministic formulation of the ensemble Kalman filter: An alternative to ensemble square root filters. *Tellus*.

Sun, D., Roth, S., and Black, M. (2010). Secrets of optical flow estimation and their principles. In *Proceedings of European Conference on Computer Vision*, pages 2432–2439.

Titaud, O., Vidard, A., Souopgui, I., and Le Dimet, F.-X. (2010). Assimilation of image sequences in numerical models. *Tellus A*, 62:30–47.