E-Infrastructures

H2020-EINFRA-2015-1

EINFRA-5-2015: Centres of Excellence for computing applications

EoCoE

Energy oriented Center of Excellence for computing applications

Grant Agreement Number: EINFRA-676629

D2.9 - M36

Paper on GHI forecasting

Project and Deliverable Information Sheet

EoCoE	Project Ref:	EINFRA-676629
	Project Title:	Energy oriented Centre of Excellence
	Project Web Site:	http://www.eocoe.eu
	Deliverable ID:	D2.9
	Lead Beneficiary:	CEA
	Contact:	Isabelle Herlin
	Contact's e-mail:	Isabelle.Herlin@inria.fr
	Deliverable Nature:	Report
	Dissemination Level:	PU*
	Contractual Date of Delivery:	M36 30/09/2018
	Actual Date of Delivery:	M36 28/09/2018
	EC Project Officer:	Carlos Morais-Pires

* - The dissemination level are indicated as follows: PU – Public, CO – Confidential, only for members of the consortium (including the Commission Services) CL – Classified, as referred to in Commission Decision 2991/844/EC.

Document Control Sheet

Document	Title :	Paper on GHI forecasting
	ID :	D2.9
	Available at:	http://www.eocoe.eu
	Software tool:	LaTeX
Authorship	Written by:	Dominique Béréziat, Isabelle Herlin
	Contributors:	Etienne Huot
	Reviewed by:	Edouard Audit, Nathalie Girard, PEC Members

Contents

List of Figures

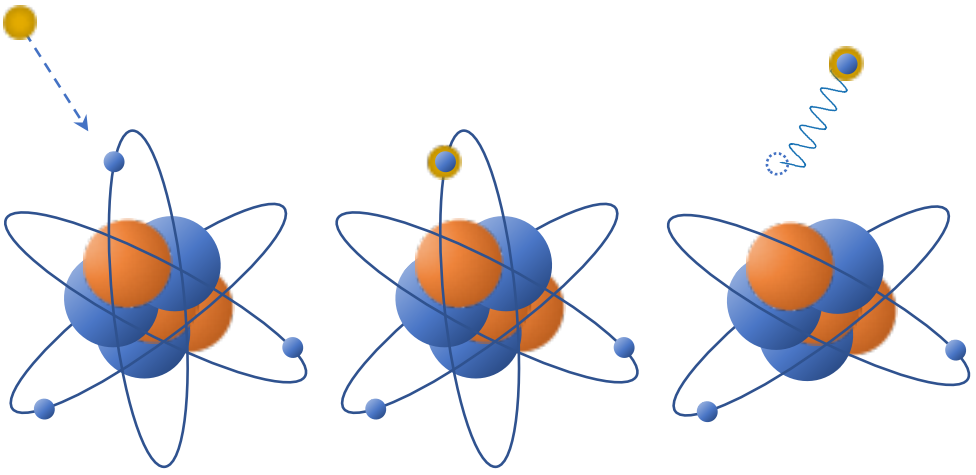
List of Tables

1. Introduction

The potential of solar energy for electricity production has been identified from the beginning of the nineteen century, for instance by the French physician Edmond Becquerel in 1839 [4]. In 1916, Robert Millikan was able to produce electricity from a photovoltaic (PV) cell. However, the yield of this cell was very poor. Based on the work of Chapin *et al.* [5], an efficient PV cell only emerged in the sixties and was considered to be an ideal tool to power satellites and other spacecrafts. Consequently, the research field of PV energy highly benefited of the raise of spatial research. Nowadays, it is possible to operationally produce electricity from solar irradiance, and, due to the fossil fuels rarefaction and ecological awareness [6, 7, 8], PV energy is viewed as an effective alternative to traditional energies.

1.1 How do solar systems produce electricity?

Two main technologies permit to produce electricity from solar irradiance. The first one relies on solar energy to warm a liquid (water) and uses - directly or indirectly - the accumulated heat to run a turbine and produce electricity in a thermal power plant. The second one is based on the photo-electric properties. After hitting an electron, a photon is caught. Consequently the electron breaks free and induces a potential difference (see Figure 1). The PV cell uses this property to produce electricity, *ie* a flow of electrons (Figure 2), in the photovoltaic power plant.



(a) a photon about to hit an electron

(b) the photon is absorbed by the electron

(c) the electron is breaking free

Figure 1: When a photon interacts with an electron.

1.2 Solar irradiance

Several solar irradiance measures are available. The Direct Normal Irradiation (DNI) is the amount of solar radiation received per surface unit that is perpendicular to the solar ray. The Diffuse Horizontal Irradiance (DHI) includes indirect sources and is the amount of radiation received per surface unit that comes from all directions, including after having been scattered by particles and molecules in the atmosphere. The Global Normal Irradiance (GHI) includes both DNI and DHI. GHI quantifies the *global* amount of shortwave radiation received from above by a unit surface on the ground: $GHI = DHI + DNI \times \cos(\theta)$ where θ is the solar zenith angle [9]. Figure 3 illustrates the significance of these measures and their complementarity.

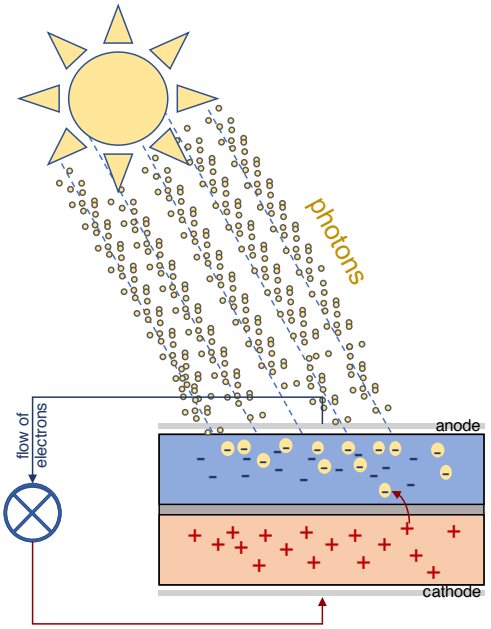


Figure 2: The photovoltaic cell principle.

1.3 Forecasting

The obvious drawback of solar energy comes from its discontinuity: seasonal, daily, nyctohemeral. Indeed, the irradiance depends on physical, meteorological and uncontrollable conditions. The availability of solar irradiance and therefore the ability to produce electricity directly depends on weather conditions. It is actually mandatory for energy providers being able to efficiently balance between this intermittent energy and traditional ones. Hence, they need to anticipate the production in order to include the solar energy amount in their network. An accurate forecast of solar irradiance is consequently a crucial concern for the management of the production network.

According to Heinemann *et al.* [10] and Diagne *et al.* [11], there are three main forecasting approaches that mainly depend on the forecast horizon.

Numerical Weather Prediction approach

The first forecasting methods appeared in the early 1980s and relied on Numerical Weather Prediction (NWP) models. These meteorological models are three-dimensional and based on primitive equations that are used to approximate the global atmospheric flow. These differential equations are the continuity equation (representing the conservation of the mass), the conservation of the momentum (describing the hydrodynamical flow) and the thermal energy conservation (linking temperature to heat sources and sinks). The primitive equations describe the evolution of the temperature, pressure and wind in the atmosphere. These NWP models are operationally used to forecast the state of atmosphere by the meteorological weather forecast centers: starting from initial conditions that are derived from observed measures, the future state of the atmosphere is by estimated from the integration in time a NWP model. Moreover, data assimilation methods [12, 13] can be applied to improve the initial conditions and consequently the forecast, thanks to observed measures and satellite images.

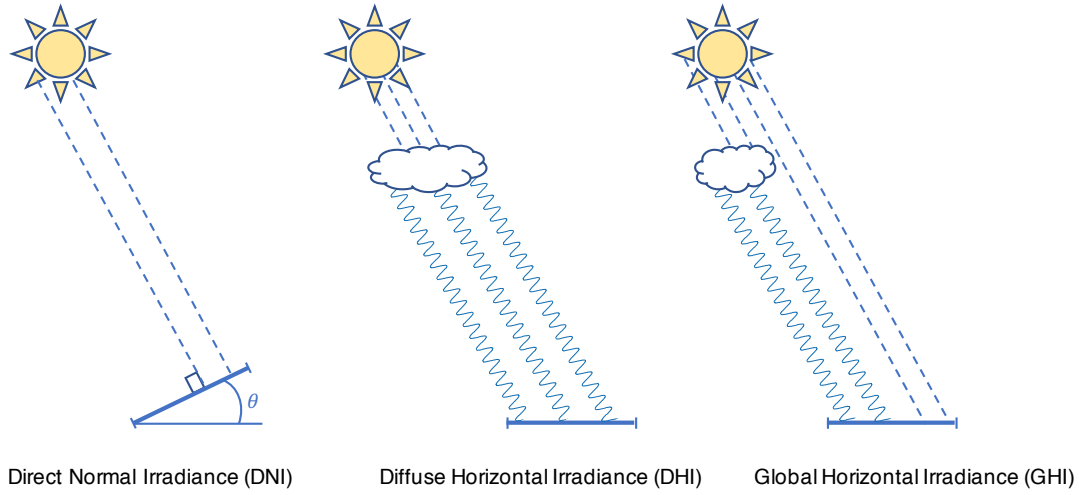


Figure 3: Measures of solar irradiance.

Macroscale NWP models, such as those applied at ECWMF [14, 15], work at a spatial resolution of 15 to 50 km and their temporal resolution ranges from 3 to 6 hours. The forecast horizon, compatible with good results, can be up to 15 days. Mesoscale models, such as MM5 from NCAR [16] and Aladin [17] from Météo-France, compute at a spatial resolution from 1 to 10 km with a temporal resolution from 30 minutes to 3 hours. These models are appropriate for day-ahead and multi-day horizon forecast, usually up to three days.

The obtention of GHI forecast using the NWP approach is based on the cloud cover forecast. A cloud cover index is computed from this forecast, on which the global irradiance is estimated, as described in Figure 4.

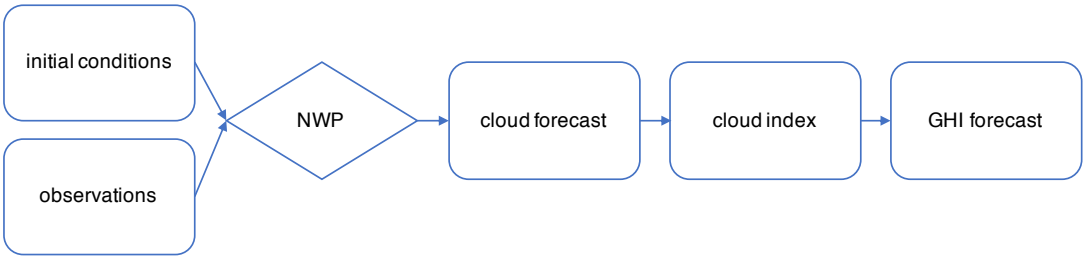


Figure 4: Forecasting approach based on NWP.

The main drawback of this approach concerns the poor resolution of the models. Even with a mesoscale model running at 1 km resolution, only an averaged cloud cover can be produced, resulting in a poor quality GHI forecast. A comparison between NWP models is proposed by Perez *et al.* in [18].

Statistical approach

The statistical approach relies on a database of the solar irradiance. Theses methods assume the existence of repetitive patterns in series of data. On a mathematical point of view, these methods rely on linear or non-linear regression techniques such as neural network [19] and auto-regressive models [20]. The forecast horizon depends on the temporal

resolution of the time series. Usually, the data are hourly averaged leading to a forecast horizon from 1 to 6 hours [21, 22, 23].

The most common reference model is the persistence model. Considering the forecast time horizon h, the method supposes that the global irradiance at time $t+h$ is best predicted by its current value at time t. GHI is then considered constant (simple persistence) or linearly evolving (smart persistence) over the temporal window t to $t+h$ (see [24] for more details). The persistence forecast accuracy decreases strongly with the forecast horizon as the cloud coverage usually changes over long time interval. The persistence model is often not correct for a forecast horizon exceeding 1 hour. However, it is extensively used as a reference for evaluating more advanced approaches.

Image-based approach

The image-based methods rely on cloud imagery coming from satellites or ground-based sky images. From a set of cloud images (at least two), a motion field is first computed. Then using the last acquired image and the estimated motion, a forecasted image is extrapolated. The cloud index and the GHI are then computed on this forecasted image [25].

One key-point of the image-based methods is the determination of motion from image data. This issue is still intensively studied in the image processing and computer vision literature. The two main techniques concern features matching between consecutive images and the so-called optical flow methods. The features-based methods rely on matching local descriptors between two images [26, 27, 28]. They determine the motion field from the displacements between these matched features [29, 30, 31]. The optical flow relies on the assumption of image brightness constancy [32]. Being an ill-posed problem (aperture problem [33, 34]), the estimation is constrained to be smooth in space and optionally in time [35, 36, 37], resulting in an estimation at pixel level.

Once the motion field is retrieved, the last image of the the sequence is advected in order to extrapolate a forecast image at the chosen temporal horizon (see Figure 5).

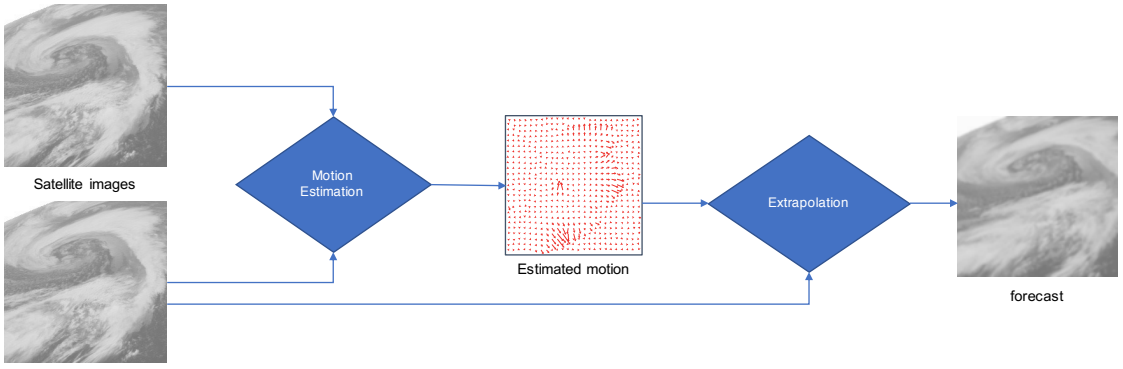


Figure 5: Forecasting scheme based on satellite images and motion estimation coupled to an extrapolation method.

This image-based approach can be applied on satellite images. For instance, MeteoSAT second generation (MSG), see Figure 6, is a geostationary satellite that provides one image every fifteen minutes with a spatial resolution of three kilometers. The last

generation of MeteoSAT, available in 2021, will deliver data with a spatial resolution of two kilometers every ten minutes [38]. Obviously, these characteristics are insufficient to achieve high temporal and spatial resolution forecasts. Consequently, satellite-based forecasts are currently inadequate for intra-day forecast.

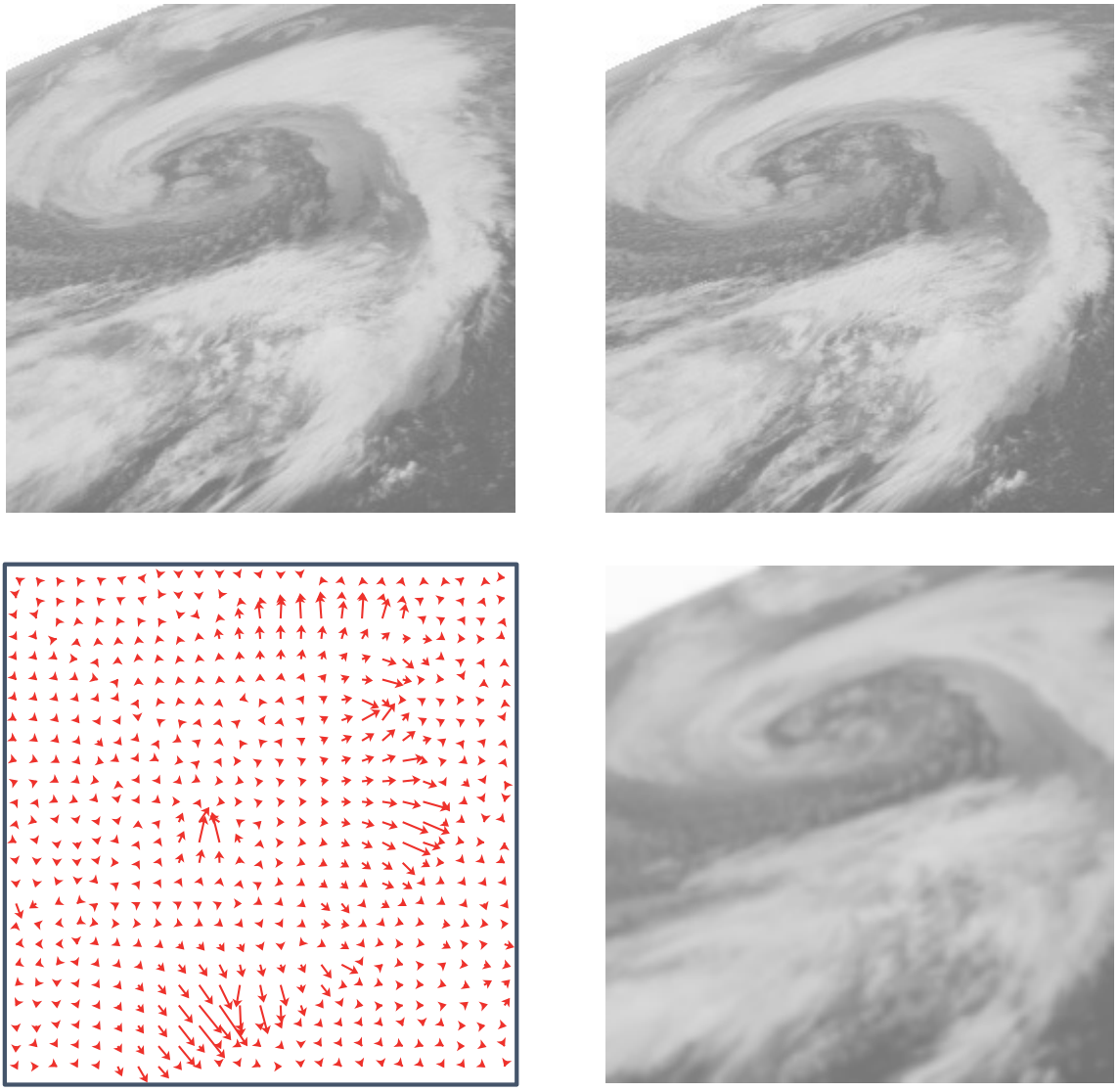


Figure 6: Top: satellite images. Bottom: motion field and forecasted image (MeteoSAT images and optical flow results from [1]).

Compared with satellite data, ground-based sky images offer a much higher spatial and temporal resolution. These sensors are hemispherical cameras taking a whole 360 degrees acquisition of the sky (see Figure 7). Their data provide an accurate description of the cloud coverage with a high temporal resolution, for instance of 10 seconds [3] for data used in this research work. However, only short forecast horizons, below 30 minutes, are reasonable with these sensors.

The main drawback of the image-based approach is that, most often, the motion estimation process does not include a physical model of the cloud coverage evolution. An alternative comes from the data assimilation technique that offers the possibility to compute an optimal solution from all types of information: evolution of the image function,

outputs of a Numerical Weather Prediction model, satellite images, ground-based images, point-wise measures. This technique also allows to combine the advantage of the pixel-based and feature-based methods as they can involve simultaneously raw observational images and characteristic features. The motion estimation process consequently includes both pixel values and clouds segmentation, resulting in a better estimation [39, 40].

In this paper, we propose an image-based approach, using ground-based sensors (fish-eye camera), and provide an optimal estimation of motion from an evolution model and all available data acquired by the fish-eye camera on a given temporal interval.

The paper is organized as follows. Section 2 describes the data used in this study. Section 3 briefly summarizes the proposed forecasting method. The optimization code techniques that are applied to obtain a real-time software are given in Section 4. Last, Section 5 displays results obtained by our method before concluding remarks in Section 6.

2. Description of Data

The approach described in the paper is tested on data acquired by a fish-eye camera (see Figure 7) developed by EDF R&D [3]. It is a low-cost alternative to standard Total Sky Imagers often used to observe the sky. The device is equipped with a Charge Coupled Device (CCD) camera and a fish-eye lens. This imager, installed in the *Piton de la Fournaise*, in the French island *La Réunion*, delivers one RGB image of the sky every ten seconds (see Figure 8). Reader is referred to [3] for technical details.

Due to the fish-eye lens, spatial structures are distorted and a geometric calibration is required to project data on a regular space grid as summarized in the following. The image obtained by the fish-eye camera results from a non-linear omnidirectional projection of the light trajectory between the scene and the lens, and a linear perspective projection of the light trajectory between the lens and the CCD sensor. The omnidirectional projection is described by a function h characterizing the lens geometry. Scaramuzza [41] approximates the function h with a Taylor development. Coefficients of this polynomial approximation are named intrinsic parameters. The perspective projection is represented by a 4×3 matrix in homogeneous coordinates, also named extrinsic parameters. The whole calibration, i.e. the estimation of intrinsic and extrinsic parameters, is processed by EDF R&D following the Scaramuzza procedure and a series of test cards as seen in Figure 9. Data is reconstructed with the Matlab Toolbox OcamCalib [42].

Figure 10 displays an example of fish-eye image and its final reconstruction on a Cartesian grid of size 512×512. In [3], it has been shown that these images allow having good performances for estimating reliable solar irradiance indices.

3. Description of the forecasting method

The forecasting method described in the paper is based on a sliding-window technique that allows to continuously process the image data and to launch a forecast, at each new acquisition, from the four last images, as seen in Figure 11. In the first step, named "Motion Estimation", the dynamics of image structures, i.e. clouds, is estimated from four consecutive acquisitions. This dynamics is described by a dense map of pixels velocity. In a second step, named "Forecasting", an extrapolation of the sky images in the future is computed up to the chosen time horizon h. The sliding-window technique recomputes the dynamics and the forecast at each new acquisition. Real-time computation (real-time refers to the temporal resolution of the acquisition process) requires delivering a new forecast every 10 seconds, which will be the objective of the optimization phase as described

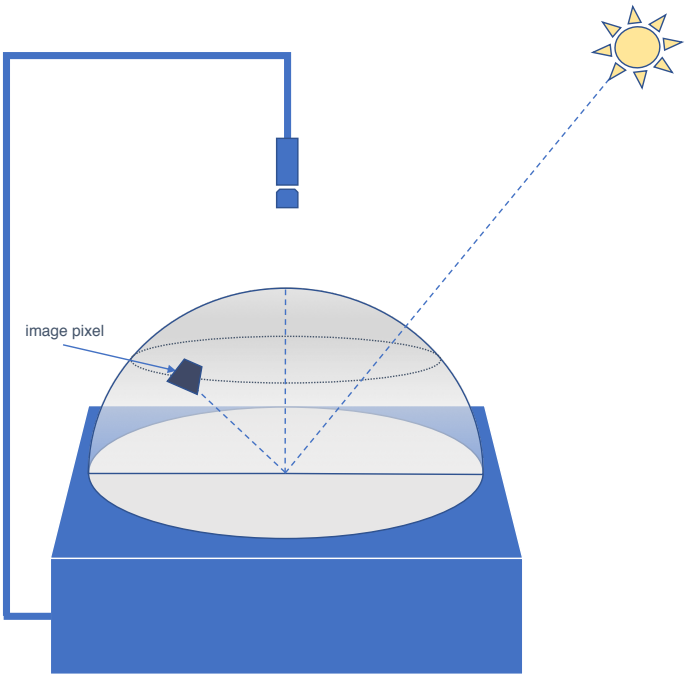


Figure 7: Fish-eye sensor acquisition, slightly inspired by [2].

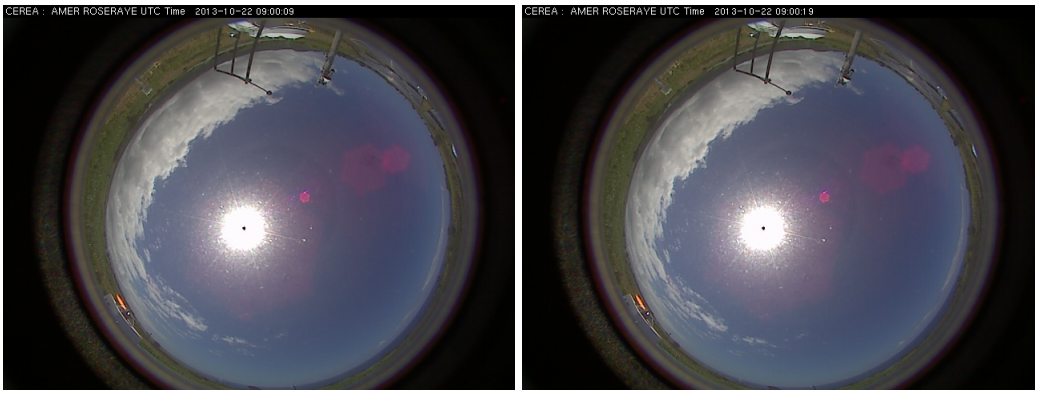


Figure 8: Two successive acquisitions (10 seconds between each acquisitions).

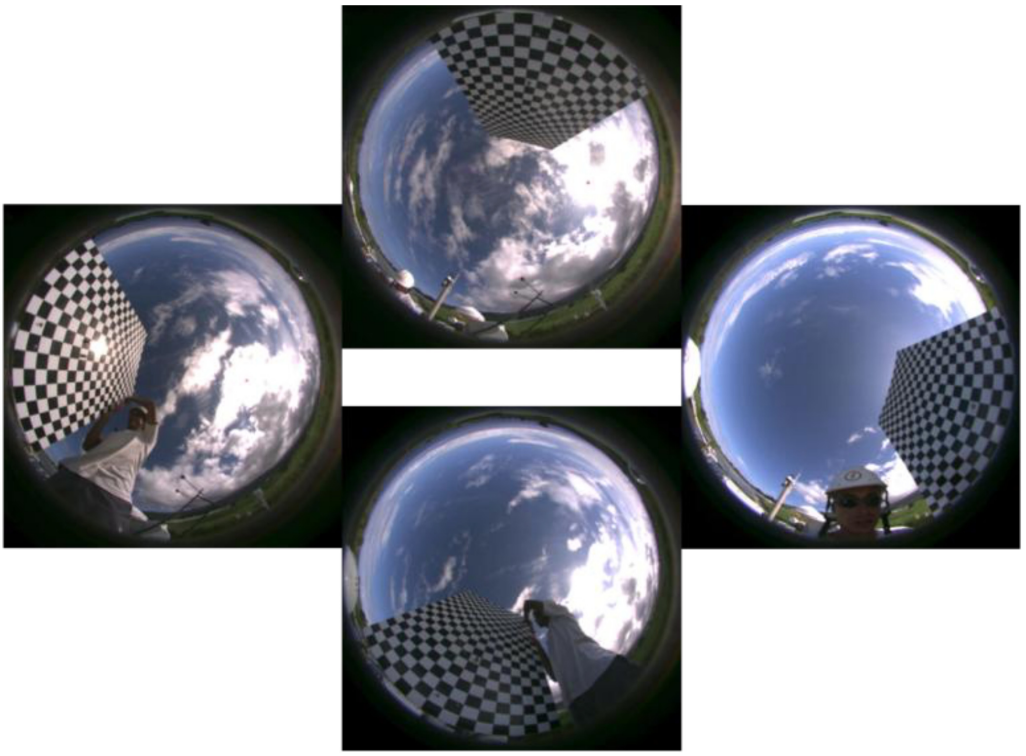


Figure 9: Chessboard used as test cards to calibrate the sky imager (from [3]).

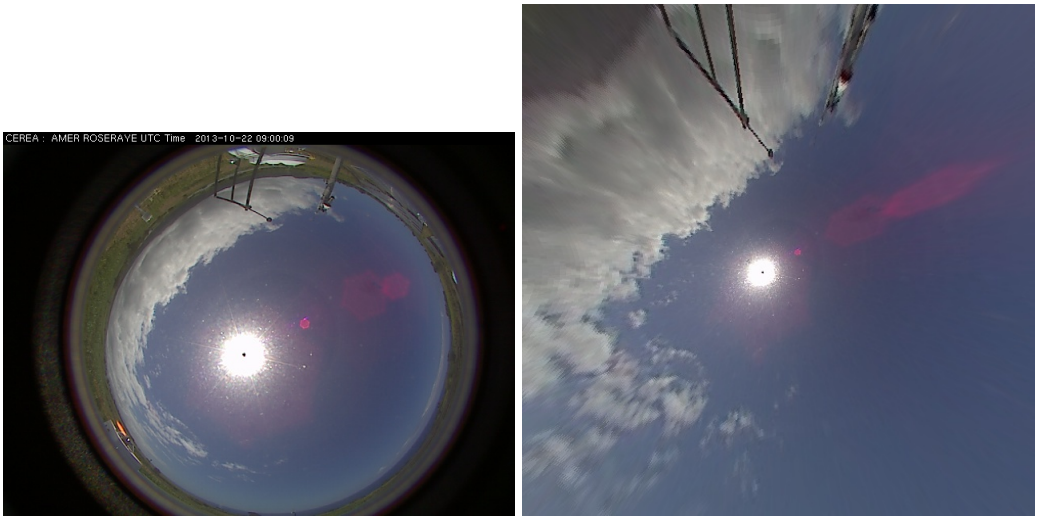


Figure 10: Left: example of data delivered by the Fish-eye sensor. Right: reconstruction on a Cartesian grid.

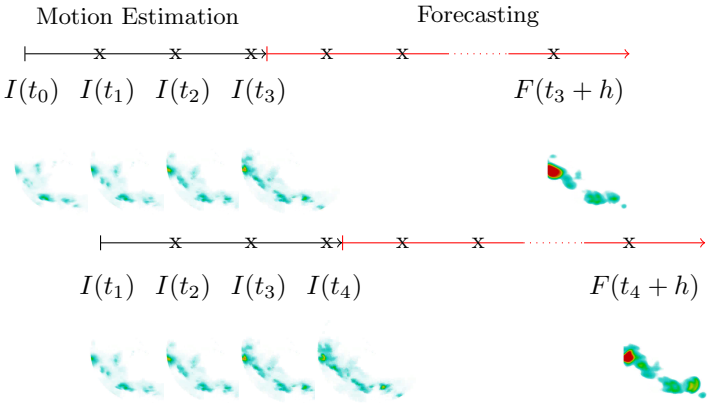


Figure 11: Illustration of the sliding-window technique. Up: from $I(t_0)$, $I(t_1)$, $I(t_2)$, $I(t_3)$, a motion field is estimated; then from $I(t_3)$ and the motion field, a forecast $F(t_3+h)$ is computed. Down: when $I(t_4)$ is available, a new motion field is estimated, then, a new forecast $F(t_4+h)$ is computed.

in Section 4.

3.1 Motion Estimation

The first step estimates the image dynamics and computes a map of velocities from the image acquisitions. Our approach relies on the literature on the optical flow methods [43]. The innovation comes from the use of a model approximating the Navier-Stokes equations, additionally to the equation on temporal evolution of images, in order to better constrain the result by physical laws. The mathematical framework issues from the data assimilation paradigm [12, 44]. Two type of observations are involved in the method:

1. images of the sky,
2. rough segmentations of the cloudy structures visible in the images.

The segmentation of clouds is computed on the sky images, combining various threshold and morphological operators. Figure 12, middle, illustrates this segmentation. At each pixel location $\mathbf{x} = (x, y)$ of the image domain Ω, we then compute $D_c(\mathbf{x})$ as the smallest distance to the clouds boundary. This results in a distance map D_c displayed on Figure 12, right. In order to make sure that our approach computes an optimal displacement according to the information coded by pixel values and that by clouds segmentation, we simultaneously rely on the image I and on the distance map D_c.

Let us briefly summarize our approach in the following and refer to [40] for complete description. We introduce the 4-components state vector as the function:

$$(\mathbf{x}, t) \mapsto \mathbf{X}(\mathbf{x}, t) = \begin{pmatrix} \mathbf{w}(\mathbf{x}, t) \\ I_s(\mathbf{x}, t) \\ \phi(\mathbf{x}, t) \end{pmatrix}. \tag{1}$$

In the following, the symbol $\mathbf{X}$ stands for the function $\mathbf{X}$ defined on the whole space-time domain and the symbol $\mathbf{X}(t_0)$ stands for the function $\mathbf{x} \mapsto \mathbf{X}(\mathbf{x}, t_0)$, t_0 being viewed as a fixed parameter. $\mathbf{X}$ is supposed integrable on Ω, the image domain, and on $[t_0, t_3]$, the temporal interval corresponding to the four images acquired at time $t_i, i = 0, 1, 2, 3$. The component $\mathbf{w} = (u, v)$ stands for velocity, I_s represents a pseudo-image, having the same characteristics than the sky image $I(\mathbf{x}, t)$, and ϕ is a signed distance map to the clouds

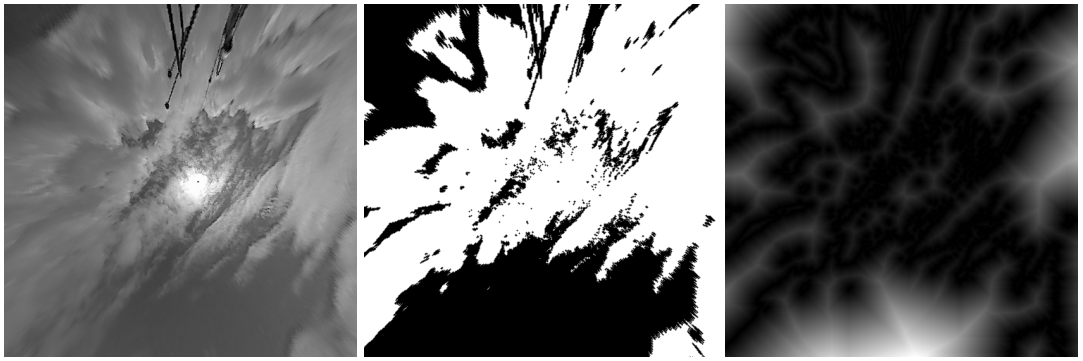


Figure 12: Example of input data: sky image (left), rough segmentation (middle), distance map to boundaries (right).

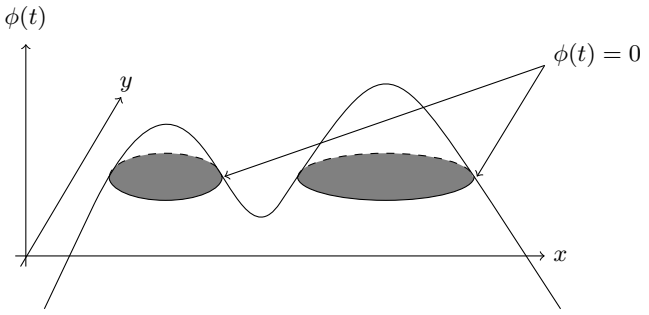


Figure 13: Implicit representation of clouds at time t: the level 0 of function $\phi(t)$ gives the clouds boundary.

boundaries displayed on I_s (see Figure 13). The sign of ϕ indicates the exterior and interior of clouds. The objective is to determine the initial value, $\mathbf{X}(t_0)$, that best fits to the data, sky images and rough segmentation, with respect to the physical evolution from t_0 to t_3. The evolution model is defined as follow:

- Quantities I_s and ϕ are transported by velocity $\mathbf{w}$. This means that image values and distance map (and consequently clouds) remain constant along pixels trajectories. This heuristics results in the following linear advection equations:

$$\frac{\partial I_s}{\partial t}(\mathbf{x},t) + \nabla I_s(\mathbf{x},t).\mathbf{w}(\mathbf{x},t) = 0 \quad (2)$$

$$\frac{\partial \phi}{\partial t}(\mathbf{x},t) + \nabla \phi(\mathbf{x},t).\mathbf{w}(\mathbf{x},t) = 0 \quad (3)$$

  where $\frac{\partial}{\partial t}$ stands for partial derivative to t and ∇ stands for the gradient operator. Equations (2) and (3) rewrite as:

$$\frac{\partial I_s}{\partial t} + \frac{\partial I_s}{\partial x}u + \frac{\partial I_s}{\partial y}u = 0 \quad (4)$$

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x}u + \frac{\partial \phi}{\partial y}u = 0 \quad (5)$$

- $\mathbf{w}$ is assumed to verify a Lagrangian constancy heuristics. This is physically consistent for a short temporal window, as it assumes that the velocity is constant along pixel trajectories. This results in the following non-linear advection equation:

$$\frac{\partial \mathbf{w}}{\partial t}(\mathbf{x},t) + \nabla \mathbf{w}(\mathbf{x},t).\mathbf{w}(\mathbf{x},t) = 0 \quad (6)$$

  $\nabla \mathbf{w}.\mathbf{w}$ is equal to $\begin{pmatrix} \frac{\partial u}{\partial x}.u + \frac{\partial u}{\partial y}.v \\ \frac{\partial v}{\partial x}.u + \frac{\partial v}{\partial y}.v \end{pmatrix}$, which includes linear and non-linear terms.

Equations (6), (2) and (3) are then summarized by:

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = 0 \tag{7}$$

where $\mathbb{M}$ is the non-linear model operator defined as:

$$\mathbb{M}(\mathbf{X}) = \begin{pmatrix} \nabla \mathbf{w}.\mathbf{w} \\ \nabla I_s.\mathbf{w} \\ \nabla \phi.\mathbf{w} \end{pmatrix} \tag{8}$$

The objective of motion estimation is to retrieve the velocity map $\mathbf{w}(t_0)$ that best transports I_s and ϕ according to the acquisitions $I(t_i)$ and distance maps $D_c(t_i)$ at times $t_i, i = 0, 1, 2, 3$. This is described by the following system:

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{x}, t) + \mathbb{M}(\mathbf{X})(\mathbf{x}, t) = 0, \ \mathbf{x} \in \Omega, \ t \in]t_0, t_3] \tag{9}$$

$$\mathbf{X}(\mathbf{x}, t_0) - \mathbf{X}_b(\mathbf{x}) = \epsilon_b(\mathbf{x}) \tag{10}$$

$$I_s(\mathbf{x}, t) - I(\mathbf{x}, t) = \epsilon_I(\mathbf{x}, t), \text{ if } t = t_i, \tag{11}$$

$$|\phi(\mathbf{x}, t)| - D_c(\mathbf{x}, t) = \epsilon_\phi(\mathbf{x}, t), \text{ if } t = t_i \tag{12}$$

- Equation (9) describes the dynamics of the state vector $\mathbf{X}$ given in Equation (7).
- Equation (10) is named the background equation. $\mathbf{X}_b$ is the background value that describes the knowledge available on the initial condition of the state vector. Its value is chosen as follows. The vector field $\mathbf{w}_b$ is taken equal to a null value or the previous estimation of $\mathbf{w}$ (on the previous acquisitions) in the sliding-window process. The image I_{sb} is taken as the first image, $I(t_0)$. The distance map ϕ_b is the signed distance map to the clouds boundaries on $I(t_0)$.
- Equation (11) quantifies the difference between the image I_s, computed by Equations (9) and (10), and the acquisition I.
- Equation (12) quantifies the difference between the signed distance map ϕ and the distance map D_c.
- ϵ_b, ϵ_I and ϵ_ϕ are unbiased Gaussian noise variables modeling uncertainties on background values, acquisitions and distance maps. They are associated to the covariance matrices B, R_I and R_ϕ.

Equations (11) and (12) are merged in an unique equation:

$$\mathbb{H}(X, I, D_c)(\mathbf{x}, t) = \epsilon_R(\mathbf{x}, t) \tag{13}$$

which $\mathbb{H} = \begin{pmatrix} I_s - I \\ |\phi| - D_c \end{pmatrix}$ being the observation operator and ϵ_R a Gaussian noise variable of covariance $R = \begin{pmatrix} R_I & 0 \\ 0 & R_\phi \end{pmatrix}$.

A variational approach, named as 4D-Var [12] in the data assimilation community, is used to compute the solution $\mathbf{X}(t_0)$, with values $\mathbf{X}(\mathbf{x}, t_0)$ at each pixel $\mathbf{x}$. A cost function J, defined by:

$$J(\mathbf{X}(t_0)) = \int_\Omega \int_{t_0}^{t_3} \frac{\epsilon_I(\mathbf{x}, t)^2}{R_I(\mathbf{x}, t)} d\mathbf{x}dt + \int_\Omega \int_{t_0}^{t_3} \frac{\epsilon_\phi(\mathbf{x}, t)^2}{R_\phi(\mathbf{x}, t)} d\mathbf{x}dt + \int_\Omega \frac{\epsilon_b(\mathbf{x})^2}{B(\mathbf{x})} d\mathbf{x} \tag{14}$$

is minimized with a steepest descent method that requires the computation of the gradient of J, ∇J. As the second and third terms of J do not only depend on $\mathbf{X}(t_0)$ but on all values $\mathbf{X}(t)$, an adjoint formulation is required for computing ∇J. Let λ be the adjoint variable, defined by:

$$\lambda(\mathbf{x}, t_3) = 0 \tag{15}$$

$$-\frac{\partial \lambda}{\partial t}(\mathbf{x}, t) + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* \lambda(\mathbf{x}, t) = \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1} \mathbb{H}(\mathbf{X}, I, D_c)(\mathbf{x}, t) \tag{16}$$

where the adjoint operator $(\frac{\partial \mathbb{M}}{\partial \mathbf{X}})^*$ is formally defined by the dual of the tangent operator: $\langle Z\eta, \lambda \rangle = \langle \eta, Z^* \lambda \rangle$. The gradient of J is given by:

$$\nabla J = B^{-1}(\mathbf{X}(0) - \mathbf{X}_b) + \lambda(t_0) \tag{17}$$

Full details on the gradient calculus may be found in [45]. Gradient of J being computed, L-BFGS, a low memory requirement and efficient quasi-Newton solver [46], is applied to perform the steepest descent. The principle of the 4D-var method is illustrated in Figure 14 and the algorithm is summarized in Algorithm 1. For a more detailed description of the motion estimation method, Reader is referred to [47].

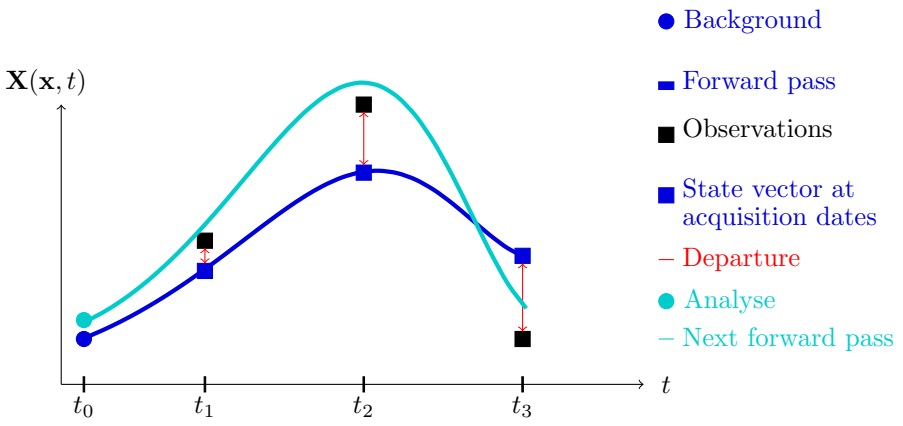


Figure 14: Principle of 4D-var: from the background value, Equation (9) is integrated. The state vector is then compared to the observations at the acquisition times with the observation operator (Departures). The analyze step computes the gradient and obtain a new value for $\mathbf{X}(t_0)$. From this new value, Equation (9) is again integrated with a resulting state vector closer to the observations than in the previous iteration.

read data ;
$\mathbf{X}(t_0) \leftarrow \mathbf{X}_b$;
while *! convergence* **do**
  Forward pass: compute $\mathbf{X}(t)$ for $t = t_0$ up to t_3, compute J ;
  Backward pass: compute $\lambda(t)$ for $t = t_3$ down to $t = t_0$, compute ∇J ;
  Optimization pass: compute the new value of $\mathbf{X}(t_0)$ from LBGFS, J and ∇J ;
end
write results ;

Algorithm 1: 4D-Var applied to Motion Estimation

3.2 Forecasting

The second step of the proposed forecasting method is the forecast at a given temporal horizon h from the currently available information. This information is based on

current and past data and includes images, distance maps and motion estimation. Rather than computing a forecast of the solar irradiance index from the current state, our approach proposes to compute a forecast F of the sky image and derive the solar irradiance index from the forecasted image.

Let us summarize the process of image forecasting. The dynamics is first estimated from the last four acquisitions as described in Subsection 3.1. This dynamics is then applied on the last image for extrapolating the position of pixels, step by step, until to the chosen horizon. This is mathematically expressed by the following advection equation:

$$\frac{\partial F}{\partial t} + \nabla F.\mathbf{w} = 0 \tag{18}$$

This equation is similar to Equation (2) and expresses the advection of F from the initial conditions (see Figure 11). The velocity map $\mathbf{w}$ used in Equation (18) is retrieved in the Motion Estimation step and denoted $\mathbf{w}_{\mathrm{ME}}$ (ME standing for Motion Estimation). The initial image $F(t_3)$ is chosen equal to the acquisition $I(t_3)$. This provides:

$$\mathbf{w}(t_3) = \mathbf{w}_{\mathrm{ME}}(t_3) \tag{19}$$

$$F(t_3) = I(t_3) \tag{20}$$

The operational use requires the highest possible value of time horizon for the forecast, in order to mitigate the electricity production. Motion should therefore not be considered as a static field. We consider the assumption of Lagrangian constancy of the velocity field, mathematically modeled by:

$$\frac{\partial \mathbf{w}}{\partial t} + \nabla \mathbf{w}.\mathbf{w} = 0 \tag{21}$$

This is similar to the equation used in Subsection 3.1, conducting to the same analysis. Equations (21) and (18) are then further integrated in time from initial conditions (19,20) up to horizon t_3+h to provide the forecasted image on which the GHI forecast is computed.

4. Implementation and code optimization

Implementation of the proposed method requires to discretize the model equations of the Forecasting step, Equations (18,21), and of the Motion Estimation step, Equations (2,3,6), and to determine the adjoint model of these last equations.

Discretization of the Forecasting step. Equations (21,18) are first approximated in time with an Euler scheme:

$$\mathbf{w}(\mathbf{x}, t+\triangle t) = \mathbf{w}(\mathbf{x}, t) - \triangle t \mathbb{M}_{\mathbf{w}}(\mathbf{w})(\mathbf{x}, t) \tag{22}$$

$$F(\mathbf{x}, t+\triangle t) = F(\mathbf{x}, t) - \triangle t \mathbb{M}_{F}(F)(\mathbf{x}, t) \tag{23}$$

$\mathbb{M}_{\mathbf{w}} = \nabla \mathbf{w}.\mathbf{w}$ and $\mathbb{M}_F = \nabla F.\mathbf{w}$ are then approximated in space with a semi-Lagrangian scheme, which is unconditionally stable and allows having a high time step value, resulting in a smaller number of integration steps. The core of the semi-Lagrangian scheme replaces the linear advection:

$$F(\mathbf{x}, t+\triangle t) = F(\mathbf{x}, t) - \triangle t \nabla F(\mathbf{x}, t).\mathbf{w}(\mathbf{x}, t) \tag{24}$$

by a non-linear warping process:

$$F(\mathbf{x}, t+\triangle t) = F(\mathbf{x} - \mathbf{w}(\mathbf{x}, t)\triangle t, t) \tag{25}$$

In Equation (25), the location $\mathbf{x} - \mathbf{w}(\mathbf{x},t)\triangle t$ rarely corresponds to a pixel on the grid. The quantity $F(\mathbf{x} - \mathbf{w}(\mathbf{x},t)\triangle t, t)$ should then be interpolated from the closest values on the Cartesian grid. This is obtained with a bilinear interpolation. Note that evaluating F at location $\mathbf{x} - \mathbf{w}(\mathbf{x},t)\triangle t$ is only an approximation. Formally, the velocity $\mathbf{w}$ should be evaluated at the previous location of $\mathbf{x}$ on its temporal trajectory, i.e. $\mathbf{x} - \mathbf{w}(\mathbf{x} - \alpha)\triangle t$. [48] estimates $\alpha = \mathbf{w}(\mathbf{x}-\alpha,t)\triangle t$ with a first order scheme in time. The approach described in this paper is based on a second order scheme named "Stable Extrapolation Two-Time Level Scheme" (SETTLS) [49], which computes α from $\alpha = (2\mathbf{w}(\mathbf{x}-\alpha,t) - \mathbf{w}(\mathbf{x}-\alpha,t-\triangle t) + \mathbf{w}(\mathbf{x},t))\frac{\triangle t}{2}$. The solution is obtained with the fix-point theorem with a limit of 5 iterations (according to experimental tests).

Discretization of Motion Estimation step. Equations are similar to those of the Forecasting step and the approximation in time and space are identical. However, the semi-Lagrangian scheme is simplified and Equation (25) is applied in order to reduce computational time. The implementation relies on a bilinear interpolation and the iterative refinement phase is suppressed. Such resulting scheme remains unconditionally stable and keeps the benefit of reducing the number of integration steps.

Adjoint model. A key point when implementing a data assimilation method is that the following property $\langle \frac{\partial \mathbb{M}}{\partial \mathbf{X}}\eta, \lambda\rangle = \langle \eta, (\frac{\partial \mathbb{M}}{\partial \mathbf{X}})^*\rangle$ of the adjoint model $(\frac{\partial \mathbb{M}}{\partial \mathbf{X}})$ should hold with the discrete model and not only for the continuous equations. Otherwise, the convergence of the algorithm is no more ensured. The adjoint model should then be determined for the discrete model and not derived as an approximation of the continuous adjoint model. To this end, we rely on an automatic differentiation tool, that allows to compute the code of the adjoint model from the code of the model. Tapenade software [50] is applied for this automatic differentiation process and produces the adjoint code used in the paper.

Optimization. The original version of the Forecasting step was a serial code running in 169 seconds on a test sequence with 3601 time steps, when computing a forecast with a time horizon of 10 minutes. The objective is to reach a run-time lower than 10 seconds, as the time resolution of the sensor is of 10 seconds. The code has first been rewritten in order to benefit from the vectorization capabilities of modern CPU. This allows to divide by 2 the run-time. Next, OpenMP statements has been introduced that allows to obtain a run-time of 4.8 seconds on 24 threads. The same techniques are used on the Motion Estimation step, but additional issues concern the adjoint model and the solver. The production of the adjoint code by Tapenade has first to be correctly tuned in order to produce a code compatible with the OpenMP statements. Second, the L-BFGS solver being a serial code, we introduced OpenMP statements in its Fortran code. The resulting optimization is well adapted to obtain a run-time compatible with the operational objectives: - before optimization, the original code of Motion Estimation ran in 68 seconds on a test sequence and on one thread, - after optimization, we obtain a run-time of 12 seconds on 24 threads. Taking advantage of the sliding-window method, it is possible to further reduce the number of iterations, and then the run-time, in the data assimilation algorithm: velocity is not initialized with a zero value, but with the one estimated on the previous window. This simple implementation technique allows to significantly decrease the number of iterations. Consequently, run-time reduces to about 2 seconds on most experimented windows.

Reader is referred to the document "Application Support: Solar Nowcasting" for details about the optimization process and study of performance.

Figure 15: 4 consecutive acquisitions, acquired on 2013-11-22 at 09:10:00.

5. Results

In this section, the potential of the proposed method is shown with an example of sky image forecasting. Figure 15 displays the four acquisitions used for computing the velocity map. These original acquisitions have been converted to gray scale images, displayed on Figure 16, as the Motion Estimation step does not use the color.

Figure 18 shows the motion result using the colored HSV representation of vectors proposed by Middlebury [51]. HSV stands for Hue, Saturation and Value. Hue codes velocity direction and saturation codes for velocity intensity. Let see Figure 17 that explains the link between color and orientation.

Figure 19 shows the forecast at several time horizons. It can be noticed that the software provides colored forecasts as the temporal integration of Equation (18) is applied on the three RGB channels. Figure 20 displays an example of forecast for a longer horizon time of 6 minutes. It exhibits some current limitations of our approach. First, the semi-

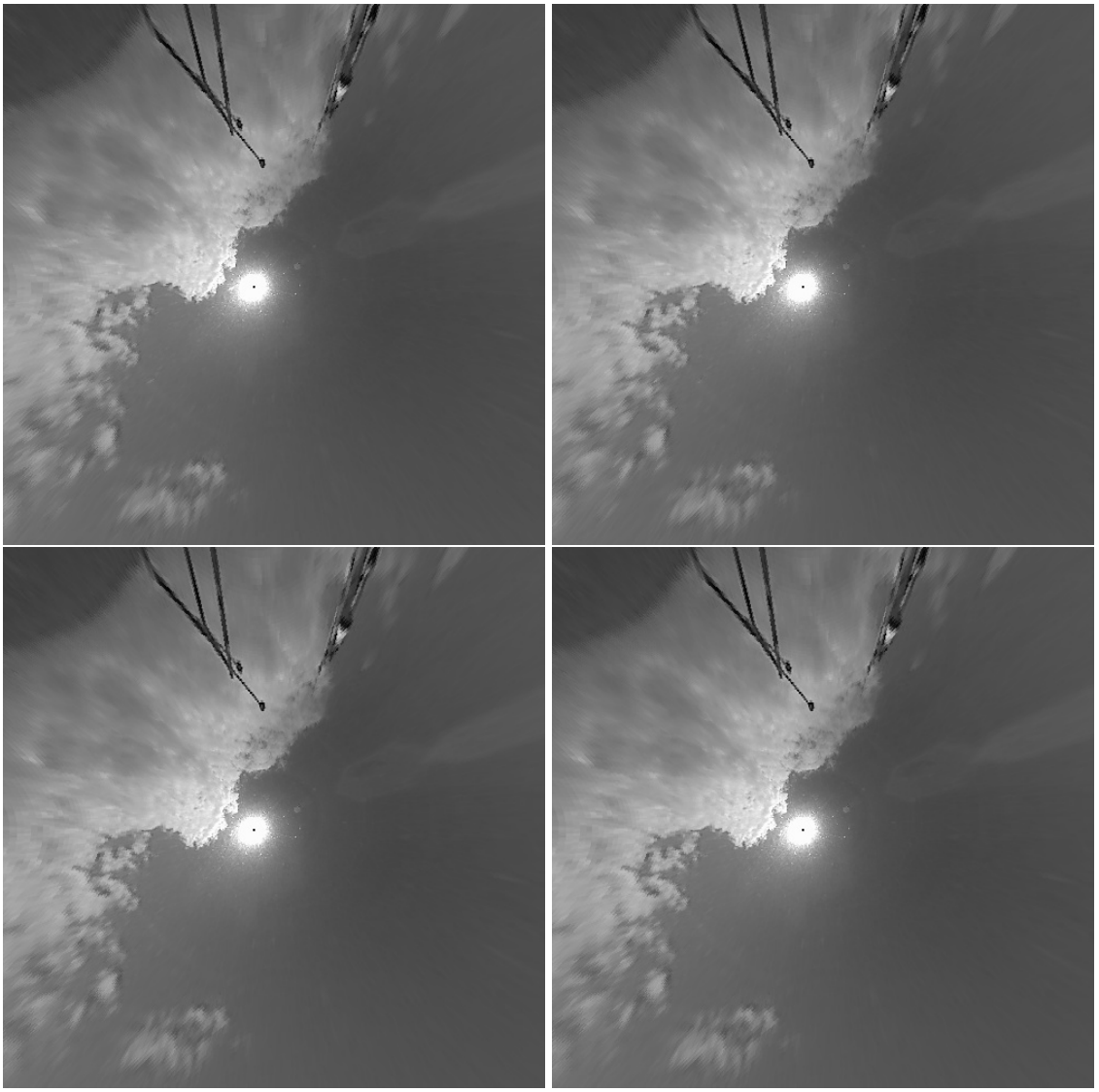


Figure 16: Fish-eye acquisitions converted to gray scale before Motion Estimation step.

Figure 17: Middlebury color representation of a vector field. Hue codes the orientation. For instance a pure red code indicates the North direction. Saturation codes the intensity: white color indicates a null velocity. Deep tint indicates the highest intensity.

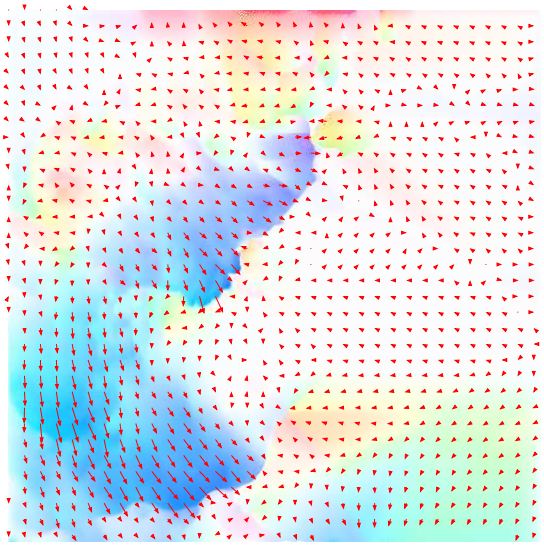


Figure 18: Velocity map using a sparse vector representation superposed to a dense colored representation.

Figure 19: Up: forecast of sky image at 10, 50, 90 and 120 seconds. Down: images acquired at the same times than forecast.

Figure 20: Left: forecast at 6 minutes. Right: corresponding acquisition.

Lagrangian scheme is not suitable for both the transport of sky image and that of velocity maps. While being unconditionally stable, the semi-Lagrangian scheme smooths the data and impacts image structures. Moreover, the Lagrangian constancy model is not valid for long time horizon. Consequently our approach is applicable for the time horizon of few minutes corresponding to the nowcast objective for which it has been designed, and can not be extended for long term forecast.

6. Conclusion

We proposed in the paper a method of nowcasting of the Solar irradiance from images delivered by a low cost Sky Imager. The method is two-fold: - in a first step, the dynamics is estimated as a dense velocity map in the image domain, from the last four observations available, - then the forecast is computed by extrapolating in time the current observation, using the velocity map. When a new observation is acquired, 10 seconds after the last one, the process is iterated with a sliding-window technique. The challenge of this nowcasting was to provide a reliable forecast in less than 10 seconds after the acquisition. This has been successfully solved as demonstrated in Section 4. The method has been illustrated in Section 5 demonstrating its potential and discussing its limitations.

The main perspective to this research is to adapt the software for longer time horizon forecasts with alternative schemes. For instance the Constrained Interpolation Profile methods [52] appear to be good candidates. Promising tests have already been done, but with a high computational cost. The proposed solution that will be implemented relies on the use of a Graphic Processor Unit (GPU) that allows a fine granularity (one thread for one pixel).

7. Acknowledgement

We would like to thank EDF R&F for providing data used in this research.

References

[1] Dominique Béréziat. *Détection et suivi de structures déformables en mouvement : application à la météorologie.* PhD thesis, Université Paris Sud, octobre 1999.

[2] P. Kuhn, B. Nouri, S. Wilbert, C. Prahl, N. Kozonek, T. Schmidt, Z. Yasser, L. Santigosa, L. Zarzalejo, A. Meyer, L. Vuilleumier, D. Heinemann, P. Blanc, and R. Pitz-Paal. Validation of an all-sky imager-based nowcasting system for industrial pv plants. *Progress in Photovoltaics: Research and Applications*, pages 1–4, November 2017.

[3] C. Gauchet, P. Blanc, B. Espinar, B. Charbonnier, and D. Demengel. Surface solar irradiance estimation with low-cost fish-eye camera. In *Workshop on Remote Sensing Measurements for Renewable Energy*, May 2012.

[4] A.E. Becquerel. Recherches sur les effets de la radiation chimique de la lumiere solaire au moyen des courants electriques. *Comptes Rendus de L'Academie des Sciences*, 9, 1839.

[5] D.M. Chapin, C.S. Fuller, and G.L. Pearson. A new silicon p-n junction photocell for converting solar radiation into electrical power. *Journal of Applied Physics*, 25:676–677, 1954.

[6] D. Boy, P. Lascoumes, C. Halpern, M. Brugidou, A. Evrard, and J. Pollard. Grenelle de l'environnement [2008 - 2011], 2017.

[7] S. Oberthür and H. Ott. *The Kyoto protocol*. Springer, Berlin [u.a.], 1999.

[8] Intergovernmental Panel on Climate Change. *Fourth Assessment Report: Climate Change 2007: The AR4 Synthesis Report*. Geneva: IPCC, 2007.

[9] Measurement GmbH Ammonit. Professional solar measurement. http://www.ammonit.com/fr/, 2017.

[10] D. Heinemann, E. Lorenz, and M. Girodo. *Solar Energy Resource Management for Electricity Generation from Local Level to Global Scale*, chapter Forecasting of solar radiation, pages 83–94. Nova Science Publishers, Inc., 2006.

[11] M. Diagne, M. David, P. Lauret, J. Boland, and N. Schmutz. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews*, 27:65–76, 2013.

[12] F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects. *Tellus*, 38A:97–110, 1986.

[13] P. Courtier, J.-N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4D-Var, using and incremental approach. 120(1367–1387), 1994.

[14] European Centre for Medium-Range Weather Forecasts, IFSDocumentation Cy 31r1. IV. Physical Processes [Online].

[15] R G Owens and Tim Hewson. Ecmwf forecast user guide. 05/2018 2018.

[16] G. A. Grell, J. Dudhia, and D. R. Stauffer. *A description of the fifth-generation Penn State/NCAR mesoscale model (MM5)*. National Center for Atmospheric Research, Boulder, Colorado, 1994.

[17] Ales Farda, Michel Déué, Samuel Somot, András Horányi, Valery Spiridonov, and Helga Tóth. Model aladin as regional climate model for central and eastern europe. 54:313–332, 05 2010.

[18] Richard Perez, Elke Lorenz, Sophie Pelland, Mark Beauharnois, Glenn Van Knowe, Karl Hemker Jr, Detlev Heinemann, Jan Remund, Stefan Müller, Wolfgang Traunmüller, Gerald Steinmaurer, D Pozo-Vazquez, Jose Ruiz-Arias, Vicente Lara Fanego, L Santigosa, Martín Gastón, and Luis Martín Pomares. Comparison of numerical weather prediction solar irradiance forecasts in the us, canada and europe. 94:305 – 326, 08 2013.

[19] P. A. Castillo, M. G. Arenas, J. J. Castillo-Valdivieso, J. J. Merelo, A. Prieto, and G. Romero. Artificial neural networks design using evolutionary algorithms. In Jose Manuel Benítez, Oscar Cordón, Frank Hoffmann, and Rajkumar Roy, editors, *Advances in Soft Computing*, pages 43–52, London, 2003. Springer London.

[20] P.J. Brockwell and R.A. Davis. *Introduction to time series and forecasting*. Springer Texts in Statistics. Springer-Verlag, New York, 2002.

[21] Hugo T.C. Pedro and Carlos F.M. Coimbra. Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy*, 86(7):2017 – 2028, 2012.

[22] Gordon Reikard. Predicting solar radiation at high resolutions: A comparison of time series forecasts. *Solar Energy*, 83(3):342 – 349, 2009.

[23] C. Voyant and G. Notton. Solar irradiation nowcasting by stochastic persistence: a new parsimonious, simple and efficient forecasting tool. *Renewable and Sustainable Energy*, 2018. In press.

[24] I. Herlin and D. Béréziat. GHI forecast results at local scale. Technical Report Deliverable D2.10, EoCoE, 2018.

[25] D. Hammer, A. dans Heinemann, C. Hoyer, R. Kuhlemann, E. Lorenz, R. Müller, and H. G. Beyer. Solar energy assessment using remote sensing technologies. *Remote Sensing of Environment*, 86:423–432, 2003.

[26] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[27] R. M. Kumar and K. Sreekumar. A survey on image feature descriptors. *Journal of Computer Science and Information Technologies*, 5(6):7668–7673, 2014.

[28] S. Krig. *Interest Point Detector and Feature Descriptor Survey*, pages 217–282. Apress, Berkeley, CA, 2014.

[29] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features, 1991.

[30] A. Bruhn and J. Weickert. Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *International Journal on Computer Vision*, 61(3):211–231, 2005.

[31] D. Sun, S. Roth, J.P. Lewis, and M. Black. Learning optical flow. In *ECCV*, pages 83–97, 2008.

[32] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Art. Int.*, 17:185–203, 1981.

[33] E. C. Hildreth. The computation of the velocity field. In *Proc. R. Soc. Lond.*, number B 221, pages 189–220, 1984.

[34] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal on Computer Vision*, 12(1):43–77, 1994.

[35] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36, 2004.

[36] F. Steinbrücker, T. Pock, and D. Cremers. Large displacement optical flow computation with warping. In *ICCV*, pages 1609–1614, 2009.

[37] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *Pat. Anal. and Mach. Int.*, (3):500–513, March 2011.

[38] R.W. Mueller, Knut-Frode Dagestad, Pierre Ineichen, Marion Schroedter Homscheidt, Sylvain Cros, Dominique Dumortier, R. Kuhlemann, Jan Olseth, G. Piernavieja, Christian Reise, Lucien Wald, and Detlev Heinnemann. Rethinking satellite based solar irradiance modelling - The SOLIS clear sky module. *Remote Sensing of Environment*, 91:160–174, 2004.

[39] Y. Lepoittevin and I. Herlin. Assimilation of radar reflectivity for rainfall nowcasting. In *IGARSS - IEEE International Geoscience and Remote Sensing Symposium*, pages 933–936, Milan, Italy, July 2015.

[40] Y. Lepoittevin, D. Béréziat, I. Herlin, and N. Mercier. Continuous tracking of structures from an image sequence. In *VISAPP - 8th International Conference on Computer Vision Theory and Applications*, pages 386–389, Barcelone, Spain, February 2013. Springer Verlag.

[41] D. Scaramuzza. *Omnidirectionnal Vision: from Calibration to Robot Motion Estimation*. PhD thesis, ETH Zurich, 2008.

[42] D. Scaramuzza. Ocamcalib toolbox: a Matlab calibration toolbox omnidirectional cameras. Google for "ocamcalib".

[43] N. Bowler, C. Pierce, and A. Seed. Development of a precipitation nowcasting algorithm based upon optical flow techniques. *Journal of Hydrology*, (288):74–91, 2004.

[44] E. Valur Hólm. Lectures notes on assimilation algorithms. Technical report, European Centre for Medium-Range Weather Forecasts Reading, U.K, April 2008.

[45] I. Herlin, D. Béréziat, N. Mercier, and S. Zhuk. Divergence-free motion estimation. In *ECCV*, volume 7575 of *LNCS*, pages 15–27, Firenze, Italy, October 2012.

[46] 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*, 23(4):550–560, 1997.

[47] Y. Lepoittevin, D. Béréziat, I. Herlin, and N. Mercier. Continuous tracking of structures from an image sequence. In *VISAPP*, pages 386–389, Barcelona, Spain, February 2013. Springer-Verlag.

[48] A. Robert. A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere-Ocean*, 1981.

[49] A. Staniforth and J. Côté. Semi-Lagrangian integration schemes for atmospheric models – a review. *Monthly Weather Review*, 119:2206–2223, 1991.

[50] L. Hascoët and V. Pascual. Tapenade 2.1 user's guide. Technical Report 0300, INRIA, 2004.

[51] S. Baker, D. Scharstein, and J.-P. Lewis. A database and evaluation methodology for optical flow. In *ICCV*, 2007.

[52] T. Yabe, F. Xiao, and T. Utsumi. The constrained interpolation profile method for multiphase analysis. *Journal of Computational Physics*, (169):556–593, 2001.