Applied Energy 288 (2021) 116656

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Multi-layer wind velocity field visualization in infrared images of clouds for solar irradiance forecasting

Guillermo Terrén-Serrano [*], Manel Martínez-Ramón

Department of Electrical and Computer Engineering, The University of New Mexico, Albuquerque, NM 87131, United States

ARTICLE INFO

Keywords:
Cloud tracking
Machine learning
Flow visualization
Beta mixture model
Sky imaging
Solar forecasting

ABSTRACT

The energy available in a solar energy powered grid is uncertain due to the weather conditions at the time of generation. Forecasting global solar irradiance could address this problem by providing the power grid with the capability of scheduling the storage and dispatch of energy. The occlusion of the Sun by clouds is the main cause of instabilities in the generation of solar energy. This investigation proposes a method to visualize the wind velocity field in sequences of longwave infrared images of clouds when there are multiple wind velocity fields in an image. This method can be used to forecast the occlusion of the Sun by clouds, providing stability in the generation of solar energy. Unsupervised learning is implemented to infer the distribution of the clouds' velocity vectors and heights in multiple wind velocity fields in an infrared image. A multi-output weighted support vector machine with flow constraints is used to extrapolate the wind velocity fields to the entire frame, visualizing the path of the clouds. The proposed method is capable of approximating the wind velocity field in a small air parcel using the velocity vectors and physical features of clouds extracted from infrared images. Assuming that the streamlines are pathlines, the visualization of the wind velocity field can be used for forecasting cloud occlusions of the Sun. This is of importance when considering ways of increasing the stability of solar energy generation.

1. Introduction

Recent legislative initiatives to stimulate the use of solar power and other sustainable energy sources will increase the number of solar power plants connected to urban power grids worldwide [1]. California aims to have 100% of clean energy generation by 2045 [2]. Similar initiatives are occurring in Japan, South Africa and the European Union, where local governments aim to generate the 24% [3], 41% [4], and 32% [5] of their energy from renewable sources by 2030 respectively. In addition, the growth of Photovoltaic (PV) solar power capacity has continued to increase in a steady exponential scale from 2000 [6].

To increase the percentage of solar energy in the electrical power grid it is important to guarantee a reliable supply of energy [7]. The forecasting of solar power provides a Smart Grid (SG) with the capability of performing energy management [8]. The interruptions in energy supply from PV systems occurs due to the projection of shadows from passing clouds [9]. Moving clouds have different effects depending on the configuration of PV arrays [10], and may cause the solar irradiance received by a PV system to increase or decrease [11]. The forecasting of solar irradiance in microgrids [12] allows automatic control of home appliances and other devices [13]. In a large-scale SG with a stable supply of energy using a mix plan of solar power from PV, concentrated solar power [14] and other systems, the forecasting of solar energy is necessary to perform an efficient management of the resources [15].

There is a documented relationship between ground measurements of direct normal irradiance and Cloud Index (CI) [16]. The relationship holds in diverse climates and weather conditions [17,18] when the CI is calculated from visible and infrared (IR) light sensors mounted in geostationary satellites [19–22]. On-ground maps of solar irradiance can be derived from the CI using geostationary satellite images [23,24].

This research aims to visualize the wind velocity field to anticipate interruptions in the supply of energy generated by PV systems [25]. The forecasting interval of this application is from 1 to 5 min [26]. This is often called nowcasting [27]. Accurate Numerical Weather Prediction (NWP) models which analyze atmospheric dynamics using satellite images are computationally expensive due to the resolution of the numerical grid necessary to analyze the forecasting intervals [28]. The variables in mesoscale meteorology models have collinearity when the objective is to forecast solar irradiation [29].

When transmitting images from a satellite, communications might have a delay of up to an hour [30]. An alternative to satellite cameras is the total sky imager [31]. This device captures sky images reflected

[*] Corresponding author.
E-mail addresses: guillermoterren@unm.edu (G. Terrén-Serrano), manel@unm.edu (M. Martínez-Ramón).

https://doi.org/10.1016/j.apenergy.2021.116656
Received 30 November 2020; Received in revised form 22 January 2021; Accepted 11 February 2021
Available online 22 February 2021

on a concave mirror, and allows a high Field Of View (FOV) [32]. This device has a number of disadvantages including its cost, and the projection of shadows on the mirror from objects in its own structure [33]. Digital cameras are less expensive and can obtain better performances around the circumsolar area with attached lenses [34]. In fact, near IR filters can attenuate the scattering produced by solar irradiance [26], and fish-eye lenses increase the FOV, which allows recording of low-cost, shadow-free sky images [35]. IR sensors are the most viable alternative when the forecast is meant for hours ahead, and includes night hours or poor daylight conditions [36]. Recent technological innovations have increased the FOV of ground-based IR images [37].

The visualization of the wind velocity field requires measurements of wind velocity at a given altitude. The wind velocity increases with the altitude in the lower atmosphere [38]. The decrease of temperature along the Troposphere can be approximated by a linear function [39]. Cloud formations are feasible in a range of altitudes that varies from the ground to the Tropopause [40]. The detection of clouds in IR images allows us to indirectly measure physical magnitudes of the wind velocity field [41]. Radiometric IR cameras provide uniform thermal imaging [42], and may be stabilized to perform atmospheric measurements [43]. In fact, microbolometer IR cameras have been used to provide statistical analysis of clouds [44] for Earth-space communication [45].

Methods of computational numerical analysis are an effective way to analyze images of clouds. The direction and magnitude of cloud velocity have been estimated applying motion vector techniques to a series of consecutive frames [46]. Through image segmentation, it is possible to identify clouds and other objects in an image [34]. The clouds' pathlines can be estimated by tracking them with a Kalman filter [35]. Classical methods of statistical modeling and linear regression have low computational requirements, and are an alternative to complex NWP models [47]. Machine learning (ML) algorithms such as artificial neural networks [48], or Support Vector Machines (SVM) [49], are promising models to find space–time correlations in cloud images.

This research utilizes two innovations. First, a Data Acquisition (DAQ) system is used for capturing radiometric long-wave IR circumsolar images combined with pyranometer measurements [50]. The DAQ is equipped with a solar tracker that updates its pan and tilt every second, maintaining the Sun in a central position in the images during the day [51]. The IR images are taken at an angle from the normal position of the camera in relation to the ground. The angle is the Sun's elevation. This causes the relative distance of a given object on the horizon to increase from top to bottom in an image. To account for this effect, a second innovation is introduced to transform the velocity vectors from the original Euclidean frame of reference to a non-linear frame of Ref. [50].

This research also proposes and implements an online ML algorithm for predicting the streamlines of multiple wind flows in an image. An unsupervised ML algorithm infers the distribution of velocity vectors and heights of multiple layers of clouds. The velocity vectors are approximated using the Weighted Lucas–Kanade (WLK) method, and are segmented and subsampled to reduce the noise of the approximation and the computational burden of the entire algorithm. A Multi-Output Weighted Support Vector Machine (ε-MO-WSVM) [52] visualizes the approximated velocity vectors to predict the trajectories of the clouds. The ε-MO-WSVM is modified from its original form adding flow constraints. The flow constraints are added so that the approximated streamlines are equivalent to the pathlines. The wind velocity field visualization can be used to forecast occlusion of the Sun by clouds, thereby predicting and preventing disruptions in the generation of energy from solar power plants.

2. Wind velocity field

The IR sensor produces a uniform thermal image. When the radiometry functionality is enabled, the pixels in a frame are turned into temperature measurements. A pixel of the camera frame is defined by a pair of euclidean coordinates $\mathbf{X} = \{(x,y)_{i,j} \mid \forall i = 1,\dots,M,\ \forall j = 1,\dots,N\}$, and the temperature of each one of the pixels is defined in Kelvin degrees as $\mathbf{T}^k = \{T_{i,j} \in \mathbb{R} \mid \forall i = 1,\dots,M,\ \forall j = 1,\dots,N\}$, where k represents a process defined as $k \in (0,\infty]$, which is a sequence of IR images ordered chronologically by time of acquisition. The temperature of a particle in the Troposphere is a function of the height [53]. The height of a pixel in a frame is approximated using the Moist Adiabatic Lapse Rate (MALR) function [50], that we define as $\phi : T \mapsto h$, knowing the temperatures obtained with the IR camera. The height of each one of the pixels in a frame are $\mathbf{H}^k = \{H_{i,j} \in \mathbb{R} \mid \forall i = 1,\dots,M,\ \forall j = 1,\dots,N\}$.

When there are multiple layers of clouds in an image, a Beta Mixture Model (BeMM) of the temperature of the pixels is expected to have multiple clusters. The number of clusters C is estimated by a previously trained detection algorithm that infers the number of wind velocity fields which are in an image. In order to infer the distribution of the temperature of the pixels with a BeMM, the temperatures are first normalized to the domain of a beta distribution such as $\bar{T}_{i,j} = [T_{i,j} - \min(\mathbf{T}^k)]/[\max(\mathbf{T}^k) - \min(\mathbf{T}^k)]$.

2.1. Beta mixture model

Consider the temperatures $\bar{T}_{i,j}$ of a given image (by omitting superindex k). The distribution of the normalized temperatures can be approximated by a mixture of beta distributions $\bar{T} \sim Be(\alpha_c, \beta_c)$ with the density function,

$$f\left(\bar{T}_{i,j};\alpha_c,\beta_c\right) = \frac{1}{\mathrm{B}\left(\alpha_c,\beta_c\right)} \cdot \bar{T}_{i,j}^{\alpha_c - 1} \cdot \left(1 - \bar{T}_{i,j}\right)^{\beta_c - 1}, \quad \alpha_c, \beta_c > 0, \tag{1}$$

where $\bar{T}_{i,j} \in (0,1)$, the beta function is $\mathrm{B}(\alpha_c,\beta_c) = [\Gamma(\alpha_c)\Gamma(\beta_c)]/[\Gamma(\alpha_c + \beta_c)]$, and the gamma function is $\Gamma(\alpha_c) = (\alpha_c - 1)!$.

The log-likelihood of the beta density function that we need to compute the expected complete data log-likelihood (CDLL) is,

$$\log p\left(\bar{T}_{i,j} \mid \alpha_c,\beta_c\right) = \left(\alpha_c - 1\right)\log \bar{T}_{i,j} + \left(\beta_c - 1\right)\log\left(1 - \bar{T}_{i,j}\right) - \log \mathrm{B}\left(\alpha_c,\beta_c\right), \tag{2}$$

and the CDLL in a mixture model is,

$$\mathcal{Q}\left(\theta^{(t)},\theta^{(t-1)}\right) = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{c=1}^{C}\gamma_{i,j,c}\log\pi_c + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{c=1}^{C}\gamma_{i,j,c}\log p\left(\bar{T}_{i,j} \mid \theta^{(t)}\right) \tag{3}$$

where $\gamma_{i,j,c} \triangleq p(y_{i,j} = c \mid \bar{T}_{i,j}, \theta^{(t-1)})$ is the responsibility of the cluster c in the sample i,j and $\theta^{(t)} = \{\alpha_c^{(t)}, \beta_c^{(t)}\}$.

The parameters in the clustering of beta distributions can be directly computed applying the Expectation Maximization (EM) algorithm [54]. In the E stage of the algorithm a prior is established and then, by using the likelihood function (3), a posterior $\gamma_{i,j,c} = p(y_{i,j} = c \mid \bar{T}_{i,j}, \theta)$ can be assigned to each sample [55]. In the M stage, the parameters α_c and β_c of each cluster that maximize the log-likelihood are computed by gradient descent of the CDLL [56]. The corresponding derivatives are,

$$\frac{\partial \mathcal{L}(\theta)}{\partial \alpha_c} = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{c=1}^{C}\gamma_{i,j,c}\frac{\partial}{\partial \alpha_c}\log p\left(\bar{T}_{i,j} \mid \alpha_c,\beta_c\right) \tag{4}$$

$$= \sum_{i=1}^{M}\sum_{j=1}^{N}\gamma_{i,j,c}\sum_{c=1}^{C}\left[\log \bar{T}_{i,j} - \psi\left(\alpha_c\right) + \psi\left(\alpha_c + \beta_c\right)\right], \tag{5}$$

$$\frac{\partial \mathcal{L}\left(\theta_c\right)}{\partial \beta_c} = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{c=1}^{C}\gamma_{i,j,c}\left[\frac{\partial}{\partial \beta_c}\log p\left(\bar{T}_{i,j} \mid \alpha_c,\beta_c\right)\right] \tag{6}$$

$$= \sum_{i=1}^{M}\sum_{j=1}^{N}\gamma_{i,j,c}\sum_{c=1}^{C}\left[\log\left(-\bar{T}_{i,j}\right) - \psi\left(\beta_k\right) + \psi\left(\alpha_c + \beta_c\right)\right]. \tag{7}$$

where $\partial \mathrm{B}(\alpha_c, \beta_c)/\partial \alpha_c = \mathrm{B}(\alpha_c, \beta_c) \cdot [\psi(\alpha_c) - \psi(\alpha_c + \beta_c)]$, and $\psi(\cdot)$ is the digamma function, which is $\psi(\alpha_c) = \Gamma'(\alpha_c)/\Gamma(\alpha_c)$.

The optimal priors are found by maximizing the CDLL with respect to π_c, constrained to $\sum_c \pi_c = 1$. As a result, the optimal priors are

$$\pi_c = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \gamma_{i,j,c}. \tag{8}$$

The cloud average heights in a frame are computed using the posterior probabilities $\gamma_{i,j,c}$ in a frame, but only in the pixels with a cloud,

$$\hat{H}_c = \frac{\sum_{i,j} \gamma_{i,j,c} \cdot H_{i,j} \cdot \mathbb{I}\left(b_{i,j} = 1\right)}{\sum_{i,j} \gamma_{i,j,c} \cdot \mathbb{I}\left(b_{i,j} = 1\right)}, \tag{9}$$

where $\mathbb{I}(\cdot)$ is the indicator function. An image segmentation algorithm indicates which pixels belong to a cloud, so that $\mathbf{B} = \{b_{i,j} \in \mathbb{B} \mid \forall i = 1, \ldots, M,\ \forall j = 1, \ldots, N\}$ is a binary image where 0 is a clear sky pixel, and 1 is a pixel belonging to a cloud [57].

The performance of a Gamma Mixture Model and a BeMM were compared to infer the distribution of the temperatures and the heights. The BeMM of the temperature of the pixels was found to be better when identifying which pixels belong to the wind velocity layer.

2.2. Motion vectors

In current computer vision literature, there are three primary methods to estimate the motion of objects in a sequence of images: Lucas–Kanade [58], Horn–Schunk [59], and Farnebäck [60]. These three methods are based on the space–time partial derivatives between two consecutive frames. Taking a different disciplinary approach, the velocity field in experimental fluid dynamics is approximated applying research methods based on signal cross-correlation operated in the frequency domain [61]. The techniques to estimate the motion vectors in an image are sensitive to the intensity gradient of the pixels. We implemented a model that removes the gradient produced by the solar direct radiation, and atmospheric scattered radiation, both of which routinely appear on the images in the course of the year. A persistent model of the outdoor germanium window of the camera removes sporadic debris that appears in the images such as water stains or dust particles [50].

A series of sequences of images with clouds flowing in different directions were simulated to cross-validate the set of parameters for each one of the mentioned methods. The investigation searched for a dense implementation of a motion vector method to approximate the dynamics of clouds. The most suitable method was found to be the Weighted Lucas–Kanade (WLK) [62], but in this application, instead of weighting the neighboring pixels with a multivariate normal distribution, the weights $\gamma_{i,j,c}$ are the posterior probabilities of the BeMM. Therefore, a pixel has a velocity vector for each cloud layer c in a frame. The optimal window size, weighted least-squares regularization, and differential kernel amplitude are: $\mathcal{W} = 16[\text{pixels}^2]$, $\tau = 1 \times 10^{-8}$, and $\sigma = 1$ respectively. The velocity components in the x-axis are $\mathbf{U}_c = \{\mathrm{u}_{i,j,c} \in \mathbb{R} \mid \forall i = 1, \ldots, M,\ \forall j = 1, \ldots, N\}$, and the velocity components in the y-axis are $\mathbf{V}_c = \{\mathrm{v}_{i,j,c} \in \mathbb{R} \mid \forall i = 1, \ldots, M,\ \forall j = 1, \ldots, N\}$. The velocity vectors are in units of pixels per frame, but knowing the geometrical transformation of the frame, they can be transformed to meters per second [50]. The geometric transformation is a function of the Sun's elevation and azimuth angles $\psi : (\varepsilon, \alpha) \mapsto \boldsymbol{\Delta}\mathbf{x}_{i,j}$ in a frame, it defines the dimensions of a pixel at a given height $\boldsymbol{\Delta}\mathbf{X} = \{(\Delta x, \Delta y)_{i,j} \mid \forall i = 1, \ldots, M,\ \forall j = 1, \ldots, N\}$. This transformation connects the x,y-axis coordinates system with the height, which is the z-axis. The relation holds even when the height is an approximation, since the components of velocity vectors are transformed with respect to the new coordinates system. The velocity vectors of each cloud layer are transformed such as,

$$u_{i,j} = \frac{\delta}{f_r} \cdot \Delta x_{i,j} \sum_{c=1}^{C} \hat{H}_c \cdot \gamma_{i,j,c} \cdot \mathrm{u}_{i,j,c} \tag{10}$$

$$v_{i,j} = \frac{\delta}{f_r} \cdot \Delta y_{i,j} \sum_{c=1}^{C} \hat{H}_c \cdot \gamma_{i,j,c} \cdot \mathrm{v}_{i,j,c} \tag{11}$$

where f_r is the frame rate of the sequence of images, and δ is velocity vectors' scale in the WLK approximation.

2.3. Velocity vectors selection

In order to approximate the potential lines and streamlines of the wind velocity field in a frame, we propose to select the most consistent velocity vectors over a sequence of consecutive frames. The main problems with this approach are that as the vectors are selected over a sequence of images, the amount of vectors is expected to be large; also when optical flow is implemented in dense manner, it yields to noisy vectors. Because of this, we threshold the velocity vectors to reduce both the algorithm's computational burden and the variance of the noise.

2.3.1. Velocity vector segmentation

The pixel intensity difference between two consecutive frames is computed to find the pixels that show a change. The pixel normalized intensities that were used to compute the velocity vectors are $\mathbf{I} = \{i_{i,j} \in \mathbb{R}^{[0,2^8)} \mid \forall i = 1, \ldots, M,\ \forall j = 1, \ldots, N\}$. The root squared intensity normalized difference is,

$$d_{i,j} = \frac{\sqrt{\left(i^{k-1}_{i,j} - i^{k}_{i,j}\right)^2}}{\sum_{i,j} \sqrt{\left(i^{k-1}_{i,j} - i^{k}_{i,j}\right)^2}}. \tag{12}$$

Matrix $\mathbf{D}$ with normalized differentials $d_{i,j}$ is vectorized and sorted from the lowest to the highest, i.e., $\mathbf{d} = \mathrm{sort}(\mathrm{vec}(\mathbf{D}))$. A vector $\mathbf{r}$ with the accumulated variance is computed as

$$r_m = \left\{ \sum_{i=1}^{m} \mathbf{d}_i \right\}_{m=1}^{N \cdot M}. \tag{13}$$

Then, vector $\mathbf{r}$ is reorganized and set in the original matrix form, defined as $\mathbf{R} = \{r_{i,j} \in \mathbb{R}^{[0,1)} \mid \forall i = 1, \ldots, M,\ \forall j = 1, \ldots, N\}$. Finally, a threshold τ is applied

$$b_{i,j} = \begin{cases} 1 & r_{i,j} \geq \tau \\ 0 & \text{Otherwise}, \end{cases} \tag{14}$$

where $\mathbf{B} \in \mathbb{B}$ is a binary image whose elements are 1 when a pixel is selected. The threshold velocity vectors in a frame k are $\mathbf{V}'^{k} = \{\mathbf{v}^{k}_{i,j} = \{u^{k}_{i,j}, v^{k}_{i,j}\} \wedge b^{k}_{i,j} = 1 \mid \forall i = 1, \ldots, M,\ \forall j = 1, \ldots, N\}$.

Based on the assumption that a cloud floating in the air follows a trajectory dictated by the wind velocity field, the wind velocity field is approximated using the segmented velocity vectors of ℓ last frames. Hence, the set of velocity vectors available to compute the wind velocity field are,

$$\tilde{\mathbf{V}}^{k} = \begin{bmatrix} \mathbf{V}'^{k} \\ \vdots \\ \mathbf{V}'^{k-\ell} \end{bmatrix} \in \mathbb{R}^{2 \times N^{k}}, \tag{15}$$

the number of samples in $\tilde{\mathbf{V}}^{k}$ is N^{k}, this number is not the same in each frame k.

2.3.2. Velocity vector and height distributions

A velocity vector $\tilde{\mathbf{v}}_i$ (by omitting superindex k) in the set $\tilde{\mathbf{V}}^{k} = \{\tilde{\mathbf{v}}^{k}_{i} \in \mathbb{R}^2 \mid \forall i = 1, \ldots, N^{k}\}$ is assumed to belong to a cloud layer c. The probability of a vector to belong to a cloud layer c is modeled as an independent normal random variable $\tilde{\mathbf{v}}_i \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$. The function of the multivariate normal distribution is,

$$p\left(\tilde{\mathbf{v}}_i \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right) = \frac{1}{\sqrt{(2\pi)^d \left|\boldsymbol{\Sigma}_c\right|}} \cdot \exp\left\{-\frac{1}{2}\left(\tilde{\mathbf{v}}_i - \boldsymbol{\mu}_c\right)^{\top} \boldsymbol{\Sigma}_c^{-1} \left(\tilde{\mathbf{v}}_i - \boldsymbol{\mu}_c\right)\right\}. \tag{16}$$

In the case when two cloud layers were detected, we propose to infer the probability distribution of velocity vectors' in each cloud layer with this model,

$$p\left(\tilde{\mathbf{v}}_i \mid \boldsymbol{\Theta}\right) \propto p\left(\tilde{\mathbf{v}}_i \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right)^{\lambda_i} \cdot p\left(\tilde{\mathbf{v}}_i \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\right)^{(1-\lambda_i)}, \tag{17}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\lambda}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\}$, and $\lambda_i \in \{0, 1\}$. $\lambda_{i,c}$ is defined as convex, considering that a velocity vector may belong to one or the other wind velocity layer, but no to both. Knowing the vectors that belong to the first cloud layer, the vectors that belong to the second cloud layer are also known, $\lambda_{i,2} = 1 - \lambda_{i,1}$. The lower bound of the data log-likelihood is found applying Jensen's inequality [63],

$$\log p\left(\tilde{\mathbf{v}}_i \mid \boldsymbol{\Theta}\right) \propto \lambda_{i,1} \cdot \log p\left(\tilde{\mathbf{v}}_i \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right) + \lambda_{i,2} \cdot \log p\left(\tilde{\mathbf{v}}_i \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\right), \tag{18}$$

so that the posterior distribution is a linear combination of the multivariate normal distributions.

The probabilistic model parameters are inferred using a fixed-point variation of the Iterated Conditional Modes (ICM) [64]. The algorithm begins by randomly assigning the velocity vectors to a cloud layer, $\lambda_{i,1} \sim \mathcal{U}(0, C-1)$. The parameters of velocity vector distributions, in Eq. (18) that maximize the data log-likelihood are computed in the first step of the algorithm. These same parameters are used to initialize the inference of the parameters of the height distributions in the second step in Eq. (22).

In the case of a multivariate normal distribution, the ICM algorithm is iteratively updates parameters. At iteration $t+1$, the means and covariances are,

$$\boldsymbol{\mu}_c^{(t+1)} = \frac{\sum_i \lambda_{i,c}^{(t)} \cdot \tilde{\mathbf{v}}_i}{\sum_i \lambda_{i,c}^{(t)}}; \quad \boldsymbol{\Sigma}_c^{(t+1)} = \frac{\sum_i \lambda_{i,c}^{(t)} \cdot \left(\tilde{\mathbf{v}}_i - \boldsymbol{\mu}_c^{(t+1)}\right)^\top \left(\mathbf{v}_i - \tilde{\boldsymbol{\mu}}_c^{(t+1)}\right)}{\sum_i \lambda_{i,c}^{(t)}} \tag{19}$$

The vectors are re-assigned to a cloud layer at the end of each parameters update, applying the maximum a posteriori (MAP) criterion

$$\lambda_{i,2}^{(t+1)} = \underset{c}{\operatorname{argmax}}\ p\left(\tilde{\mathbf{v}}_i \mid \boldsymbol{\mu}_c^{(t+1)}, \boldsymbol{\Sigma}_c^{(t+1)}\right) - 1 \tag{20}$$

$$\lambda_{i,1}^{(t+1)} = 1 - \lambda_{i,2}^{(t+1)}. \tag{21}$$

After completing the inference of the velocity vectors distribution, it is possible to infer the cloud layer's height using the same method. The velocity vectors in an image were calculated using the WLK method. The algorithm approximates the velocity vector using a set of pixels inside a window. The result is that the velocity vectors do not exactly correspond to a clouds' pixels, which are in motion. Instead, the velocity vectors are assigned to nearby pixels. To identify which layer of clouds c, is the highest and which one is the lowest, the height distribution of the pixels is inferred using the MAP classification of the velocity vectors $\mathbf{v}'_{i,j}$ in a image.

The height of the pixels within the cloud are modeled as independently distributed normal random variables $H_{i,j} \sim \mathcal{N}(\mu_c, \sigma_c^2)$. The probabilistic model to infer the distribution of heights of each cloud layer in a frame is,

$$\log p\left(H_{i,j} \mid \boldsymbol{\Theta}\right) \propto \rho_{i,j,1} \cdot \log p\left(H_{i,j} \mid \mu_1, \sigma_1^2\right) + \rho_{i,j,2} \cdot \log p\left(H_{i,j} \mid \mu_2, \sigma_2^2\right), \tag{22}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{P}, \mu_1, \sigma_1, \mu_2, \sigma_2\}$, and $\rho_{i,j,c} \in \{0, 1\}$ is a convex variable so that $\rho_{i,j,2} = 1 - \rho_{i,j,1}$.

The ICM algorithm is also used to the infer the parameters of the height distributions model. The $\rho_{i,j,c}$ are initialized to the MAP classification of the velocity vectors $\mathbf{v}'_{i,j}$ using the parameters that were inferred using all the velocity vectors $\tilde{\mathbf{v}}_i$ in Eq. (15),

$$\rho_{i,j,2} = \underset{c}{\operatorname{argmax}}\ p\left(\mathbf{v}'_{i,j} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right) - 1 \tag{23}$$

$$\rho_{i,j,1} = 1 - \rho_{i,j,2}. \tag{24}$$

The parameters of the height distributions are updated using the formulas in Eq. (19). The algorithm eventually converges so that the pixels in the frame are segmented where a cloud appears. The segmentation is performed according to the MAP classification of height distributions,

$$\rho_{i,j,2}^{(t+1)} = \underset{c}{\operatorname{argmax}}\ p\left(H_{i,j} \mid \mu_c^{(t+1)}, \sigma_c^{2(t+1)}\right) - 1 \tag{25}$$

$$\rho_{i,j,1}^{(t+1)} = 1 - \rho_{i,j,2}^{(t+1)}. \tag{26}$$

In order to find the height of a given cloud layer, the heights are averaged with this formula,

$$\bar{H}_c = \frac{\sum_{i,j} \rho_{i,j,c} \cdot H_{i,j} \cdot \mathbb{I}\left(b_{i,j} = 1\right)}{\sum_{i,j} \rho_{i,j,c} \cdot \mathbb{I}\left(b_{i,j} = 1\right)}. \tag{27}$$

The wind velocity fields are organized into upper and lower layers by average height $\bar{H}_c$. In this way, each detected wind velocity field has a distribution of velocity vectors, and another distribution of heights.

2.3.3. Sampling

In order to reduce the computational burden of the algorithm, a subset of the velocity vectors is selected according to the estimated probability distributions of the vectors. At each layer c, we define the importance weights $w_{i,c}^k$ as

$$w_{i,c}^k \triangleq p\left(\tilde{\mathbf{v}}_i^k \middle| \theta_c\right), \quad w_{i,c}^k \in \mathbb{R}^+. \tag{28}$$

The weights are normalized to have the characteristics of a probability mass function such as $\sum_{i=1}^{N_k} \hat{w}_{i,c}^k = 1$.

The Cumulative Probability Function (CDF) is computed as

$$\tilde{w}_{i,c}^k = \left\{ \sum_{m=1}^{i} \hat{w}_{m,c}^k \right\}_{i=1}^{N^k}. \tag{29}$$

In order to select samples for each distribution $p(\tilde{\mathbf{v}}_i^k \mid \boldsymbol{\theta}_c)$, N^*/C samples are drawn from a uniform distribution,

$$z_{j,c}^k \sim \mathcal{U}\left(0, 1\right), \quad j = 1, \ldots, \frac{N^*}{C}, \tag{30}$$

For each value $z_{j,c}^k$, a sample is selected with the criterion

$$I_{j,c}^k = \operatorname{argmin}\ \mid \tilde{w}_{i,c}^k - z_{j,c}^k \mid, \quad \forall i = 1, \ldots, N_k \quad \forall j = 1, \ldots, \frac{N^*}{C}. \tag{31}$$

The selected vectors are the ones whose CDF is closest to the values of the uniform samples $z_{j,c}^k$,

$$\tilde{\mathbf{v}}_c^{*k} \triangleq \tilde{\mathbf{v}}_{I_{j,c}^k}^k, \quad \forall j = 1, \ldots, \frac{N^*}{C}. \tag{32}$$

The subset of selected velocity vectors in frame k for the cloud layer c is $\mathbf{V}_c^{*k} = \{(\tilde{u}, \tilde{v})_{j,c}^{*k} \in \mathbb{R}^2 \mid \forall j = 1, \ldots, N^*/C\}$, the subset of Euclidean coordinate pairs of those selected vectors is $\mathbf{X}_c^{*k} = \{(x, y)_{j,c}^{*k} \in \mathbb{N}^2 \mid \forall j = 1, \ldots, N^*/C\}$.

Assuming that the prior is uniform, the posterior probabilities are,

$$z_{i,c}^{*k} \triangleq \frac{p\left(\tilde{\mathbf{v}}_i^{*k} \middle| \theta_c\right)}{\sum_{c=1}^{C} p\left(\tilde{\mathbf{v}}_i^{*k} \middle| \theta_c\right)}. \tag{33}$$

The sampling is repeated for as many cloud layers detected. All selected subsets of vectors, coordinate pairs, and weights form the dataset that is used to approximate the wind velocity field,

$$\mathbf{X}^{*k} = \begin{bmatrix} x_1^{*k} & y_1^{*k} \\ \vdots & \vdots \\ x_{N^*}^{*k} & y_{N^*}^{*k} \end{bmatrix}, \ \mathbf{V}^{*k} = \begin{bmatrix} \tilde{v}_1^{*k} & \tilde{u}_1^{*k} \\ \vdots & \vdots \\ \tilde{v}_{N^*}^{*k} & \tilde{u}_{N^*}^{*k} \end{bmatrix}, \ \mathbf{Z}^{*k} = \begin{bmatrix} z_{1,1}^{*k} & z_{1,c}^{*k} \\ \vdots & \vdots \\ z_{N^*,1}^{*k} & z_{N^*,c}^{*k} \end{bmatrix}, \tag{34}$$

where $N^k \gg N^*$.

3. Flow visualization

The atmosphere is a system where the dynamics are continuously changing [28]. This fact implies that a wind velocity field exists, but we can only visualize it where clouds are present. From ground level to the Tropopause, the wind flow can have multiple layers with different velocities in each one of the layers. The wind flow may also be convective, however, for the sake of simplicity, we assume that the multi-layer flow which is observed in IR images is a multi-layer laminar flow. This analysis neither considers the z-component in the motion of a cloud (which is not observable) nor the possible inter-crossing of cloud layers.

3.1. Wind velocity field estimation

Three methods were implemented to estimate the extrapolation function and compare their performances. The first method uses a weighted ε-support vector regression machine (ε-WSVM) for each one of the velocity components. The second method is a ε-MO-WSVM that estimates both velocity components. The third is an innovation which uses a ε-MO-WSVM with flow constraints (ε-MO-WSVM-FC) to estimate both velocity components. The flow constraints are used to force the extrapolated wind flow to have zero divergence or curl, so it can be assumed that, in the approximated wind flow, streamlines are equivalent to the cloud pathlines.

The regression problem can be formulated as the optimization of a function with the form,

$$f\left(\mathbf{x}_i\right) = \mathbf{w}^\top \varphi\left(\mathbf{x}_i\right) + b, \quad \forall i = 1,\dots,N^*, \quad \mathbf{w}, \mathbf{x}_i \in \mathbb{R}^D, \ b \in \mathbb{R}. \tag{35}$$

where $\mathbf{x}_i \triangleq \mathbf{x}_i^{*k}$ in our problem, and where $\varphi(\cdot)$ is a transformation into a higher dimensional (possibly infinite) Hilbert space $\mathcal{H}$ endowed with a dot product $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle$. A function $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ is a dot product if it is a bivariate positive semi-definite function that maps $\mathbf{x}_i, \mathbf{x}_j$ into $\mathbb{R}$, commonly called a Mercer's kernel or simply a kernel function.

3.1.1. Support vector machine for regression

Assuming $\mathbf{v}_i = \{u_i, v_i\} \triangleq \mathbf{v}_i^{*k}$, the regression problem in a ε-SVM is formulated with an ε-insensitive loss function, which penalizes the errors $|\varepsilon| > 0$ [65] for each one of the components in $\mathbf{v}_i$ and for each cloud layer c as

$$\left| u_i - f\left(\mathbf{x}_i\right) \right|_\varepsilon = \max\left[0, \left| u_i - f\left(\mathbf{x}_i\right) \right| - \varepsilon \right], \quad \forall i = 1,\dots,N, \quad u_i, \varepsilon \in \mathbb{R}, \tag{36}$$

and identically for v_i. The ε-insensitive loss function can be seen as a tube of radius ε adjusted around the regression hyper-plane via the Support Vectors (SV) [66].

The samples are weighted by their probability of belonging to wind velocity field c [67],

$$z_i \triangleq z_i^{*k}, \quad z_i \in \mathbb{R}^{\leq 1}. \tag{37}$$

$$c_i = z_i \cdot \frac{C}{N} \tag{38}$$

The L2-norm and ε-loss function is applied to the model weights,

$$\min_{\mathbf{w}, b, \xi, \xi^*} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{N} \sum_{i=1}^{N} z_i \left(\xi_i + \xi_i^* \right) \tag{39}$$

$$\text{s.t.} \begin{cases} u_i - \mathbf{w}^\top \varphi\left(\mathbf{x}_i\right) - b & \leq \varepsilon + \xi_i \\ \mathbf{w}^\top \varphi\left(\mathbf{x}_i\right) + b - u_i & \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases} \quad i = 1,\dots,N, \tag{40}$$

and identically for v_i. The slack variables ξ_i were introduced to relax the constraints of the optimization problem and to deal with unfeasible optimization problems [68]. The primal objective function aims to find the trade off between the regularization term ε, the allowed errors or slack variables ξ_i and ξ_i^*, and the complexity of the model c_i per weighted sample.

The proposed kernel functions in this analysis are,

$$\mathcal{K}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{x}_i^\top \mathbf{x}_j, \tag{41}$$

$$\mathcal{K}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\gamma \cdot \|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \tag{42}$$

$$\mathcal{K}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left(\gamma \cdot \mathbf{x}_i^\top \mathbf{x}_j + \beta\right)^d, \tag{43}$$

where $\gamma,\ \beta \in \mathbb{R}$, and $d \in \mathbb{N}$ are the kernel hyperparameters that need cross-validation. [66]. These kernel functions are referred to as linear, radial basis function (RBF) or square exponential, and polynomial of order d respectively [69].

In order to optimize the constrained problem in functional (39) and constraints (40) a Lagrangian functional is constructed with the functional and the set of constraints through a dual set of new variables [70], which leads to a solvable Quadratic Programming problem (QP). The Lagrangian functional is

$$\mathcal{L}\left(\mathbf{w}, b, \alpha, \alpha^*, \beta, \xi, \xi^*, \varepsilon, \eta, \eta^*\right) = \tag{44}$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 + c_i \sum_{i=1}^{N} \left(\xi_i + \xi_i^*\right) \ \dots \tag{45}$$

$$- \sum_{i=1}^{N} \left(\eta_i \xi_i + \eta_i^* \xi_i \ *\right) - \sum_{i=1}^{N} \alpha_i \left(\varepsilon + \xi_i - u_i + \mathbf{w}^\top \varphi\left(\mathbf{x}_i\right) + b\right) \ \dots \tag{46}$$

$$- \sum_{i=1}^{N} \alpha_i^* \left(\varepsilon + \xi_i^* + u_i - \mathbf{w}^\top \varphi\left(\mathbf{x}_i\right) - b\right), \quad \forall i = 1,\dots,N, \quad \eta_i, \eta_i^* \in \mathbb{R}. \tag{47}$$

The derivatives of the primal variables $\mathbf{w}, \varepsilon, \xi_i, \xi_i^*$ yield to the following set of equations, which is a case of Karush-Kuhn–Tucker (KKT) conditions,

$$\mathbf{w}^\top = \sum_{i=1}^{N} \left(\alpha_i^* - \alpha_i\right) \varphi\left(\mathbf{x}_i\right), \tag{48}$$

$$0 = \sum_{i=1}^{N} \left(\alpha_i - \alpha_i^*\right), \tag{49}$$

$$0 = c_i - \alpha_i - \eta_i, \tag{50}$$

$$0 = c_i - \alpha_i^* - \eta_i^*. \tag{51}$$

These conditions, together with the complimentary KKT conditions (which force the product of dual parameters α_i, α_i^* with the constraints to be zero) leads to the following dual functional by substitution on the Lagrangian:

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \quad \frac{1}{2} \cdot \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right)^\top \mathbf{K} \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right) + \sum_{i=1}^{N} \left(\alpha_i - \alpha_i^*\right) u_i + \varepsilon \cdot \mathbf{1}^\top \left(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*\right) \tag{52}$$

$$\text{s.t.} \begin{cases} \mathbf{1}^\top \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right) = 0 \\ 0 \leq \alpha_i, \alpha_i^* \leq c_i \end{cases} \quad \forall i = 1,\dots,N. \tag{53}$$

where $\mathbf{1}_{1 \times N} = [1,\dots,1]^\top$, and matrix $\mathbf{K}$ is a Gram matrix of dot product such that $\mathbf{K}_{i,j} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. The minimal of the primal function is equivalent to the saddle point on the Lagrangian formulation. The approximated function is,

$$f\left(\mathbf{x}\right) = \sum_{i=1}^{N} \left(\alpha_i - \alpha_i^*\right) \cdot \mathcal{K}\left(\mathbf{x}_i, \mathbf{x}_i\right) + b, \tag{54}$$

where b is obtained from the complimentary KKT conditions.

3.1.2. Multi-output weighted support vector machine

When the wind velocity field function is approximated by ε-MO-SVM, the primal regression can be formulated as

$$\mathbf{v}_i = \mathbf{W}^\top \varphi(\mathbf{x}_i) + \mathbf{b}, \tag{55}$$

where each one of the column vectors of primal parameter matrix $\mathbf{W}$ approximates one of the velocities in vector $\mathbf{y}_i$. Primal parameters are a function of the dual parameters as well, but the dual parameters $\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_i^*$ are vectors in a 2-dimensional multi-output problem.

Since independent variables are represented in vectors $\mathbf{v}_i$, the training set is defined in a vector $\tilde{\mathbf{v}}_{1 \times 2N}$, and so are the dual parameters $\tilde{\boldsymbol{\alpha}}_{1 \times 2N}$ and $\tilde{\boldsymbol{\alpha}}^*_{1 \times 2N}$ for notation simplicity.

The gram matrix of dot products between input patterns $\varphi(\mathbf{x}_i)$ can be interpreted as the covariance matrix between variables $\mathbf{v}_i$. Indeed

$$\begin{aligned}\mathbb{E}\left((\mathbf{v}_i-\mathbf{b}^\top)(\mathbf{v}_j-\mathbf{b})\right) &= \mathbb{E}\left(\mathbf{W}^\top\varphi(\mathbf{x}_i)\varphi(\mathbf{x}_j)^\top\mathbf{W}\right)\\ &= \begin{pmatrix}\varphi(\mathbf{x}_i)^\top\Sigma_{11}\varphi(\mathbf{x}_j) & \varphi(\mathbf{x}_i)^\top\Sigma_{12}\varphi(\mathbf{x}_j)\\ \varphi(\mathbf{x}_i)^\top\Sigma_{21}\varphi(\mathbf{x}_j) & \varphi(\mathbf{x}_i)^\top\Sigma_{22}\varphi(\mathbf{x}_j)\end{pmatrix},\end{aligned} \tag{56}$$

where the 2 × 2 covariance $\mathbb{E}\left(\mathbf{W}\mathbf{W}^\top\right)$ is interpreted as a model for the dependencies between elements in $\mathbf{v}_i$, i.e.

$$\mathbb{E}\left(\mathbf{W}\mathbf{W}^\top\right) = \begin{pmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}. \tag{57}$$

If we consider that both vertical and horizontal velocities are independent, then $\Sigma_{12} = \Sigma_{21} = \mathbf{0}$. If we assume further that $\Sigma_{11} = \Sigma_{22} = \mathbf{I}$ for simplicity, which, in turn leads to

$$\mathbb{E}\left((\mathbf{v}_i-\mathbf{b}^\top)(\mathbf{v}_j-\mathbf{b})\right) = \begin{pmatrix}\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j) & 0\\ 0 & \mathcal{K}(\mathbf{x}_i,\mathbf{x}_j)\end{pmatrix}. \tag{58}$$

The Gram matrix $\tilde{\mathbf{K}}_{DN\times DN}$ in the ε-MO-SVM formulation for independent components is,

$$\tilde{\mathbf{K}} = \begin{pmatrix}\mathbf{K} & \mathbf{0}\\ \mathbf{0} & \mathbf{K}\end{pmatrix}. \tag{59}$$

The full kernel matrix in a ε-MO-WSVM is computationally expensive, and it is not implemented in this research.

The extension of weights in the ε-MO-WSVM requires weighting each sample in each output [71],

$$\tilde{z}_i = \left[z\ldots z_N\ z_1\ldots z_N\right]^\top, \quad \sum_{i=1}^{D\cdot N}\tilde{z}_i = 2, \quad \tilde{z}_i\in\mathbb{R}^{\leq 1}. \tag{60}$$

The dual formulation of the QP problem for the ε-MO-WSVM is,

$$\min_{\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\alpha}}^*}\ \frac{1}{2}\cdot\left(\tilde{\boldsymbol{\alpha}}-\tilde{\boldsymbol{\alpha}}^*\right)^\top\tilde{\mathbf{K}}\left(\tilde{\boldsymbol{\alpha}}-\tilde{\boldsymbol{\alpha}}^*\right)+\tilde{\mathbf{y}}^\top\left(\tilde{\boldsymbol{\alpha}}-\tilde{\boldsymbol{\alpha}}^*\right)+\varepsilon\cdot\mathbf{1}^\top\left(\tilde{\boldsymbol{\alpha}}+\tilde{\boldsymbol{\alpha}}^*\right) \tag{61}$$

$$\text{s.t.}\begin{cases}\mathbf{1}^\top\left(\tilde{\boldsymbol{\alpha}}-\tilde{\boldsymbol{\alpha}}^*\right)=0\\ \mathbf{0}\leq\tilde{\alpha}_i,\tilde{\alpha}_i^*\leq\tilde{c}_i\end{cases}\quad\forall i=1,\ldots,2N, \tag{62}$$

where the extended weighted complexity is $\tilde{c}_i = \tilde{z}_i/2N$.

3.1.3. Multi-output weighted support vector machine with flow constraints

Assuming that the analyzed air parcel is sufficiently small so that the flow can be considered approximately incompressible and irrotational, a new set of flow constraints are added to the original set of constraints with the purpose of visualizing the wind velocity field to force the divergence and the vorticity to zero:

$$\text{s.t.}\begin{cases}\left(\tilde{\mathbf{v}}_c^{k\top}\mathbf{\Delta}_{x,y}\dot{\mathbf{V}}\right)\cdot\left(\tilde{\mathbf{v}}_c^{k\top}\mathbf{\Delta}_{x,y}\dot{\mathbf{V}}\right)^\top &= 0\\ \left(\tilde{\mathbf{v}}_c^{k\top}\mathbf{\Delta}_{x,y}\dot{\mathbf{D}}\right)\cdot\left(\tilde{\mathbf{v}}_c^{k\top}\mathbf{\Delta}_{x,y}\dot{\mathbf{D}}\right)^\top &= 0,\end{cases} \tag{63}$$

where the matrices of this expression are defined in Eqs. (64) (65) and (66). To compute the vorticity and divergence, the differentiation of the velocity field along the x-axis, the and y-axis is implemented using operator

$$\mathbf{\Delta}_{x,y} = \begin{bmatrix}\boldsymbol{\Delta}_x & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Delta}_y\end{bmatrix}_{2N\times 2N}, \tag{64}$$

where the differential operators $\boldsymbol{\Delta}_x$ and $\boldsymbol{\Delta}_y$ are defined as,

$$\boldsymbol{\Delta}_x = \begin{bmatrix}-1 & 0 & \ldots & 0\\ 1 & -1 & \ddots & \vdots\\ 0 & 1 & \ddots & 0\\ \vdots & \ddots & \ddots & -1\\ 0 & \ldots & 0 & 1\end{bmatrix}_{N\times N} \quad ; \quad \boldsymbol{\Delta}_y = \begin{bmatrix}-1 & \ldots & 0\\ \vdots & \ddots & \vdots\\ 0 & \ldots & -1\\ 1 & \ldots & 0\\ \vdots & \ddots & \vdots\\ 0 & \ldots & 1\end{bmatrix}_{N\times N}. \tag{65}$$

The operators of the velocity field's vorticity and divergence are respectively,

$$\dot{\mathbf{V}} = \begin{bmatrix}1 & \ldots & 0\\ \vdots & \ddots & \vdots\\ 0 & \ldots & 1\\ 1 & \ldots & 0\\ \vdots & \ddots & \vdots\\ 0 & \ldots & 1\end{bmatrix}_{2N\times N} \quad ; \quad \dot{\mathbf{D}} = \begin{bmatrix}1 & \ldots & 0\\ \vdots & \ddots & \vdots\\ 0 & \ldots & 1\\ -1 & \ldots & 0\\ \vdots & \ddots & \vdots\\ 0 & \ldots & -1\end{bmatrix}_{2N\times N}. \tag{66}$$

The velocity field is extrapolated to the entire frame using the inferred parameters in frame k

$$\hat{\mathbf{v}}_c^k = \left(\tilde{\boldsymbol{\alpha}}_c^k - \tilde{\boldsymbol{\alpha}}_c^{*k}\right)\cdot\mathcal{K}\left(\mathbf{X}^{*k},\mathbf{X}\right)+\mathbf{b}_c^k, \tag{67}$$

where the velocity components are $\hat{\mathbf{U}}_c^k \triangleq \hat{\mathbf{v}}_{x,c}^k$, and $\hat{\mathbf{V}}_c^k \triangleq \hat{\mathbf{v}}_{y,c}^k$, where $\hat{\mathbf{U}}_c^k,\hat{\mathbf{V}}_c^k\in\mathbb{R}^{M\times N}$.

To compute the flow constraints, the velocity field has to be extrapolated to the whole frame using Eq. (67). The constraints in Eq. (63) are added to the constraints in the dual formulation of the ε-MO-SVM in Eq. (61).

4. Wind velocity field dynamics estimation

To estimate the wind velocity field dynamics, velocity vectors are approximated using the WLK method. The velocity vectors are weighted by the posterior probabilities of the cloud layers in the image, and transformed to the cloud layer plane. The velocity vectors are segmented and sampled to reduce the noise and the computation burden. The parameters of the ε-MO-WSVM-FC are cross-validated, and the model is trained to estimate the wind velocity field of the detected cloud layers. If an optimal set of parameters is available, it is possible to proceed with the training of ε-MO-WSVM-FM without performing the cross-validation. After training the ε-MO-WSVM-FC, the testing is performed to extrapolate the wind velocity field to the whole image. The streamlines are computed using Eq. (68) to visualize the trajectory of a cloud. The potential lines are not shown in the Figs. 5(a)–6(f), but they are computed with Eq. (69).

The physical process of cloud formation and evolution over time is part of atmospheric thermodynamics and may have divergence and vorticity [72]. The air parcel in one frame is very small compared to the whole volume of air contained in the atmosphere. Within this frame we consider it feasible that there is no vorticity or divergence, and the approximated streamlines are equivalent to the pathlines. Henceforth, the obtained results are a numerical approximation of the actual atmospheric air parcel flow.

When we assume that a flow does not have divergence and vorticity, the stream and velocity potential functions are orthogonal, and we can apply Cauchy–Riemann equations to calculate their rates of change [73]. For the stream function we determine $d\phi = v_x dy - v_y dx$ using samples of functions. The trapezoidal rule of numerical analysis is applied to solve the definite integrals [74]. The values of a streamline are,

$$\boldsymbol{\Phi}_c = \frac{\hat{H}_c}{2}\left[\left\{\sum_{m=1}^{i}\hat{\mathbf{u}}_{m,c}\odot\Delta\mathbf{y}_{m,c}\right\}_{i=1}^{N} - \left\{\sum_{m=1}^{j}\hat{\mathbf{v}}_{m,c}\odot\Delta\mathbf{x}_{m,c}\right\}_{j=1}^{N}\right], \tag{68}$$

where $\odot$ denotes the element-wise matrix multiplication. This is the sum of element-wise products between each velocity component and its opposite differential increments.

The total change in the potential function is $d\psi = v_x dx + v_y dy$, so we can determine the potential in each pixel $\mathbf{x} = \{x, y\}$ as,

$$\boldsymbol{\Psi}_c = \frac{\hat{H}_c}{2}\left[\left\{\sum_{m=1}^{j}\hat{\mathbf{u}}_{m,c}\odot\Delta\mathbf{x}_{m,c}\right\}_{j=1}^{N} + \left\{\sum_{m=1}^{i}\hat{\mathbf{v}}_{m,c}\odot\Delta\mathbf{y}_{m,c}\right\}_{i=1}^{M}\right]. \tag{69}$$

the sum of each element-wise product between the velocity components, and their differential increment.

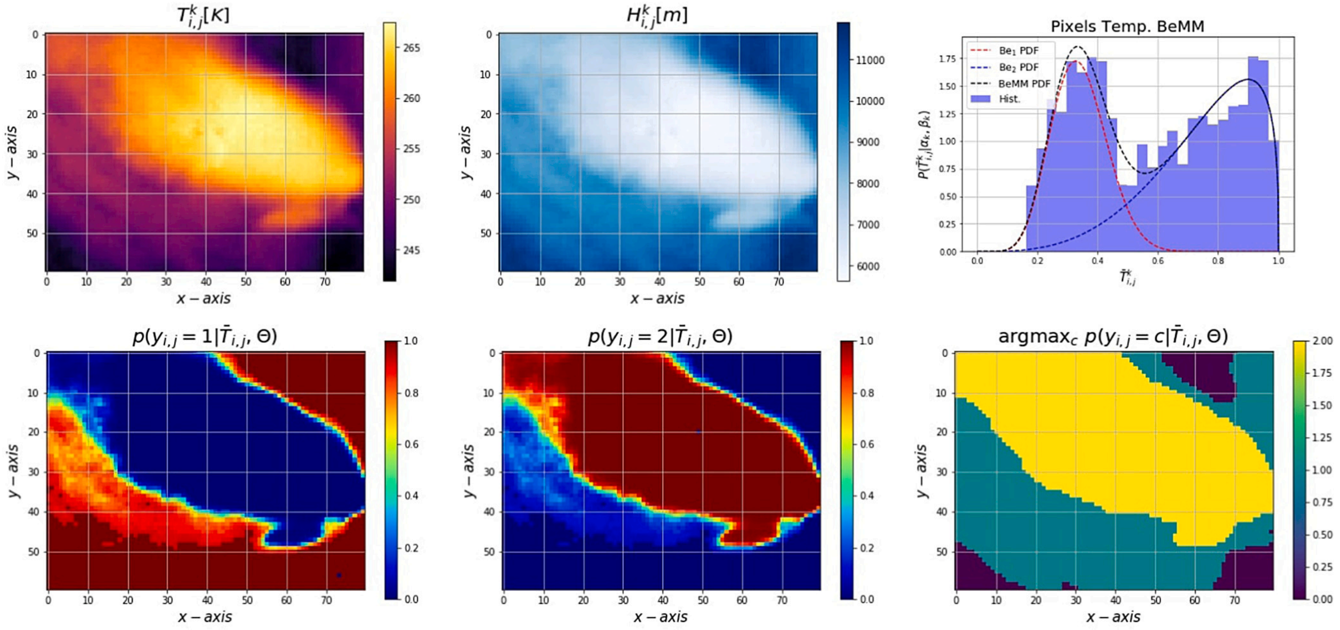


Fig. 1. The images in the first row from left to right show the pixel temperatures, pixel heights and the BeMM distribution of the normalized pixel temperatures. The images in the middle row show the posterior probabilities of the upper layer, the posterior probabilities of the lower layer, and MAP classification of the pixels.

5. Experiments

To infer the wind velocity field this method utilizes data acquired by a system that captures circumsolar IR images and measures global solar irradiance using a pyranometer. The IR sensor is a Lepton[1] radiometric camera, which has a wavelength from 8 to 14 μm and provides a uniform thermal image. When the radiometric functionality is enabled, the pixels in a frame are turned into temperature measurements in centikelvin units. The resolution of an IR image is 80 × 60 pixels, and the diagonal FOV is 60°. The data is publicly accessible in the Dryad repository [75].

The weather features that were used to compute cloud height as well as to remove cyclostationary artifacts [50] on the IR images are: atmospheric pressure, air temperature, dew point and humidity. The weather station is set to measure every 10 min, so the data was interpolated to match the IR images' sampling interval. The weather station is located at the University of New Mexico Hospital, and both its real-time and historical data are publicly accessible.[2]

The images in the top row of Fig. 1 show the temperature of the pixels obtained using the radiometric functionality of the IR camera in Kelvin (left pane), the height of the pixels in meters after applying the MALR transformation to the temperatures (center pane), and the temperature $\bar{T}_{i,j}$ histogram in light blue (right pane). Beta distributions in Eq. (1), are in red and blue, and the BeMM result in Eq. (3) is in black. The images in the bottom row show the temperature posterior probabilities $\gamma_{i,j,1}$ of the upper layer (left), the temperature posterior probabilities $\gamma_{i,j,2}$ of the lower layer (center), and the MAP classification of the pixels (right). In the image that shows the MAP classification of the pixels, those in dark blue are the segmented pixels that do not belong to a cloud ($b_{i,j} = 0$).

5.1. Training data construction

To create a data set for validation purposes we selected 21 consecutive images, per day, on six different days. The images were selected due to the presence of different types of clouds distributed across different heights. The selected images were captured during different seasons and different times of the day. The images from three of the six days show a layer of cirrustratus in winter in the morning, altostratus in spring in the afternoon and stratocumulus in summer in the afternoon. The other three days show two layers of altostratus and stratocumulus in winter at noon, cirrustratus and altocumulus in spring in the afternoon, and cirruscumulus and cumulus in summer in the morning.

The proposed algorithm only requires the validation dataset to be labeled as it is an unsupervised online machine learning algorithm. The validation dataset is used to find the optimal parameters of the algorithm which segments and subsamples the velocity vectors from the last 6 consecutive frames.

The targets of the ε-WSVM are the velocity vectors computed using the WLK method. The machine is cross-validated and trained for each frame using the selected velocity vectors of the last 6 frames. The testing error is that of the ε-WSVM approximating the WLK velocity vectors.

The average height, velocity magnitude and angle of a cloud was manually calculated for each cloud layer in each sequence of images to use them as ground truth. To do this, the pathline intercepting the Sun was manually segmented. The distance that a cloud has moved in the pathline was calculated in each frame. The height of a cloud layer was measured by segmenting the clouds along the sequence of images. The calculated height, velocity magnitude and angle of each cloud layer was averaged across the validation sequences of images.

The wind velocity is a relative measure of the actual velocity in a frame. The algorithm does not need the actual wind velocity. The algorithm requires the height of the clouds to define the space of camera's FOV. The velocity vectors are transformed from pixels per frame to meters per second. Each pixel is projected to its actual dimension in the space defined by the camera's FOV. To know the distance that a cloud will travel in a given time to occlude the Sun, the magnitude of the projected velocity vectors in the space defined by the camera's FOV is calculated. The relative measure of the wind velocity vectors (in pixels per frame) and the height of the clouds (in meters) are connected together in the frames by the geometric transformation. For this reason, the wind velocity that it is required is not the actual wind velocity but the relative wind velocity measured in the frame.

[1] https://www.flir.com/

[2] https://www.wunderground.com/dashboard/pws/KNMALBUQ473

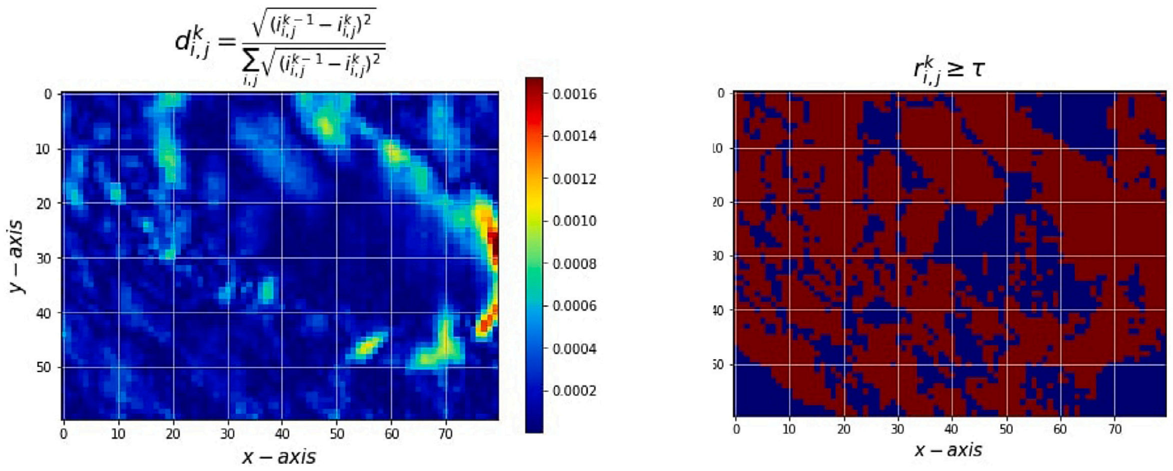


Fig. 2. The figures illustrate the proposed method to discriminate between points which probably show a moving cloud air parcel and those that probably show a pixel without movement. The left graph shows the computation of the squared difference between two consecutive frames. The right graph shows in blue those pixels which are considered not moving as their squared difference value is less than a previously validated threshold τ.

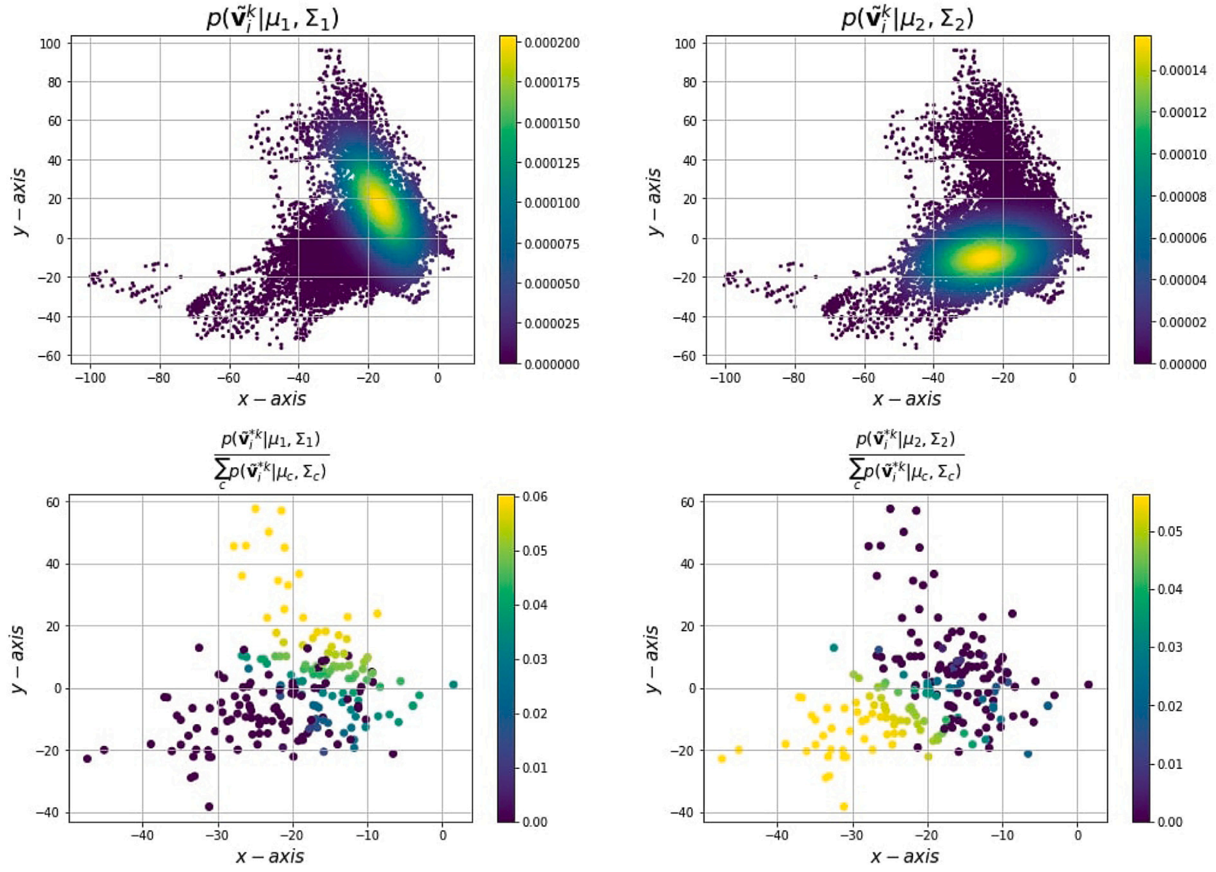


Fig. 3. Probability distribution of the velocity vectors and the subsampling implemented to decrease the computational cost. The upper row shows the distribution of the measured velocities represented in $\mathbb{R}^2$. The colormap represents the likelihood of the velocities conditional to a point belonging to the lower layer of clouds (left) and upper layer (right). The lower row shows the downsampled set of vectors (and their posterior probabilities) after applying the downsampling methodology.

5.2. Velocity vectors calculation, segmentation and subsampling parameters validation

The parameters δ in Eq. (10), τ in Eq. (14), ℓ in Eq. (15) of the velocity vectors selection algorithm were validated using the six sequences of images described above. The velocity estimator was ε-WSVM with a linear kernel. The parameters of ε-WSVM, ε and C, were cross-validated in the same process. The flow velocity constraints were not applied in the validation.

The average of the approximated wind velocity field height, magnitude and angle was computed, and the Mean Absolute Percentage Error (MAPE) was calculated between the measured and the averaged approximation in each frame. The MAPE was averaged and differentiated across consecutive frames. The combination of parameters that had less averaged MAPE plus total difference of MAPE between consecutive frames was selected. This added difference of MAPE was used to account for the accuracy of the selected model parameters, but also the stability of the models. This stability is optimal if good results are consistently obtained for each one of the images in the same sequence.

The optimal amplitude δ of the velocity vector in Eq. (10), was found to be $\delta = 2.29$. The optimal threshold τ in the segmentation of the velocity vector in Eq. (14), was found to be $\tau = 0.95$. The optimal number of velocity vectors from ℓ last frames to form the dataset in Eq. (15), was found to be $\ell = 6$. The optimal number of selected samples N^* by sampling algorithm in Eq. (34), was found that $N^* = 200$ samples are sufficient to perform a robust extrapolation of the wind velocity field to the entire frame.

The velocity vectors that were segmented in a frame with two layers of clouds are shown in Fig. 2. The velocity vectors from the last 6 frames after applying the segmentation are shown in the upper row of Fig. 3. In this figure, the colors represent the likelihoods conditional to the upper cloud layer (left), and lower cloud layer (right). The sampled velocity vectors in a frame with two layers of clouds are shown in the bottom row. Fig. 4 shows the selected velocities in the bottom row of Fig. 3 in their corresponding coordinates. In this figure, the colors represent the posterior probabilities conditional to the upper cloud layer (left), and lower cloud layer (right).

5.3. ε-MO-WSVM-FC parameters validation

After optimal values of δ, τ and ℓ have been chosen, the parameters of the proposed ε-MO-WSVM-FC are cross-validated using the validation data or an online ML approach. This means that the experiments with the ε-MO-WSVM-FC have two different setups. In the first, the parameters are cross-validated in each testing frame. In the second, the parameters are fixed to the optimal values obtained in a previous cross-validation using the validation data. This is done to check for the validity of the previously obtained parameters, which would significantly reduce the velocity field estimation computational burden.

The objective of the constraints is that the divergence and vorticity of approximated velocity field are zero in the clouds' plane. The velocity fields shown in Figs. 5(b) to 6(f) have divergence and vorticity after the field is projected to the camera plane. This is caused by the non-linear geometric transformation applied to the velocity vectors.

5.4. Wind velocity field estimation with new data

Unlike the ε-MO-WSVM-FC, the experiments with the ε-WSVM and ε-MO-WSVM use only the first setup. These models are validated and trained for each testing frame. The results are compared with a Gaussian process for regression (GPR) for each one of the velocity components [76], a Multi-Output Ridge Regression (MO-RR) with independent components (which is a special case of Tikhonov's regularization [77]) and a multi-output Gaussian process for regression (MO-GPR) with correlation between velocity components [78].

The testing data is composed of sequences of 28 images from 10 different days. The sequences are from different seasons and different times of the day. Five days have one velocity field layer and the other five have two layers. 75% of this data is chosen for the online training and validation of the models. The rest of data is used for testing. The testing set is from $k + \ell$ frames ahead of the training set from frame k. The number of frames ahead is equal to the lag of the velocity vectors in the data $\ell = 6$. The methodologies implemented in the validation are the standard grid search and 3-fold cross-validation. The parameters cross-validated in the ε-WSVM, ε-MO-WSVM and ε-MO-WSVM-FC are C and ε. The MO-RR requires the cross-validation of the regularization parameter. In the GPR and MO-GPR the parameters are found by numerical gradient, optimizing the marginal log-likelihood. The kernel functions are: linear, RBF, polynomial of order two ($\mathcal{P}^2$), and polynomial of order three ($\mathcal{P}^3$). The optimal parameters for the ε-MO-WSVM-FC are displayed in Table 2.

Table 1
The table above shows the testing results of the different kernel learning methods without flow constraints. The first method is the ε-WSVM-FC, the second is ε-MO-WSVM, the third is the GPR, the fourth is the MO-RR and the fifth is the MO-GPR. The wind velocity fields approximated by all the methods have low MAE and WMAE with high divergence and vorticity. The fastest methods are GPR and MO-GPR as the optimization of the parameters is performed via numerical gradient.

$\mathcal{K}(\mathbf{x}, \mathbf{x}^*)$	MAE	WMAE	$\nabla \cdot \vec{V}$	$\nabla \times \vec{V}$	Time [s]
ε-WSVM					
Linear	13.37	12.55	$1.69 \cdot 10^3$	$2.17 \cdot 10^3$	90.01
RBF	13.39	12.61	$6.25 \cdot 10^3$	$6.40 \cdot 10^3$	365.72
$\mathcal{P}^2$	14.06	13.22	$1.20 \cdot 10^4$	$1.19 \cdot 10^4$	2413.79
$\mathcal{P}^3$	14.90	13.95	$8.98 \cdot 10^4$	$9.48 \cdot 10^4$	3468.75
ε-MO-WSVM					
Linear	13.27	**12.49**	$1.30 \cdot 10^3$	$1.35 \cdot 10^3$	162.70
RBF	14.00	13.13	$\mathbf{1.21 \cdot 10^3}$	$\mathbf{1.22 \cdot 10^3}$	560.54
$\mathcal{P}^2$	14.25	13.53	$1.43 \cdot 10^4$	$1.71 \cdot 10^4$	5635.31
$\mathcal{P}^3$	19.29	18.12	$8.89 \cdot 10^5$	$8.92 \cdot 10^5$	7284.54
GPR					
Linear	12.56	12.56	$2.62 \cdot 10^3$	$3.27 \cdot 10^3$	6.50
RBF	12.89	12.88	$1.24 \cdot 10^4$	$1.27 \cdot 10^4$	6.43
$\mathcal{P}^2$	12.52	12.50	$7.27 \cdot 10^3$	$9.02 \cdot 10^3$	6.44
$\mathcal{P}^3$	12.67	12.68	$2.72 \cdot 10^4$	$3.11 \cdot 10^4$	**6.42**
MO-RR					
Linear	12.62	12.58	$2.62 \cdot 10^3$	$3.31 \cdot 10^3$	6.71
RBF	13.43	13.35	$3.95 \cdot 10^3$	$7.24 \cdot 10^3$	11.80
$\mathcal{P}^2$	12.55	12.55	$1.53 \cdot 10^4$	$1.15 \cdot 10^4$	29.76
$\mathcal{P}^3$	12.70	12.64	$3.17 \cdot 10^5$	$2.20 \cdot 10^5$	41.16
MO-GPR					
Linear	12.57	12.58	$2.69 \cdot 10^3$	$3.34 \cdot 10^3$	8.07
RBF	12.81	12.80	$1.21 \cdot 10^4$	$1.23 \cdot 10^4$	17.67
$\mathcal{P}^2$	12.53	12.55	$1.10 \cdot 10^4$	$1.09 \cdot 10^4$	11.31
$\mathcal{P}^3$	12.54	12.55	$4.12 \cdot 10^4$	$4.49 \cdot 10^4$	11.19

The criteria for selecting the most suitable model and kernel function for our application is a trade-off between divergence and vorticity, Weighted Mean Absolute Error (WMAE), and the computing time. The values of these metrics are summarized for the models without constraints in Table 1, and for the ε-MO-WSVM-FC in Table 2. The experiment of the ε-MO-WSVM without flow constraints using a $\mathcal{P}^3$ kernel is shown in Fig. 5(a), and that same experiment implemented with the ε-MO-WSVM-FC using a linear kernel is in Fig. 5(b). In sequences of images in which two layers of clouds were detected, the experiments of the ε-MO-WSVM-FC using a linear kernel to approximate wind velocity field are shown in Figs. 6(a) to 6(f).

The experiments were carried out in the Wheeler high performance computer of UNM-CARC, which uses SGI AltixXE Xeon X5550 at 2.67 GHz with 6 GB of RAM memory per core, 8 cores per node, 304 nodes total, and runs at 25 theoretical peak FLOPS. It has Linux CentOS 7 installed. The DAQ is localized on the roof of UNM-ME building in Albuquerque, NM.

6. Discussion

This investigation adds new insights into the computational methods to forecast the trajectory of clouds and predict the occlusion of the Sun. The proposed method visualizes the wind velocity field using IR images of clouds. The algorithm introduced here differs from previous investigations in that it is based on fluid dynamics. The experiments show that the pathlines are equivalent to the streamlines in images where is possible to extract enough information about the wind flow from the clouds.

From the summary of the experiments presented in Tables 1 and 2, several aspects can be highlighted. The overall performance of the ε-SVM increases when the samples are weighted, since the weights

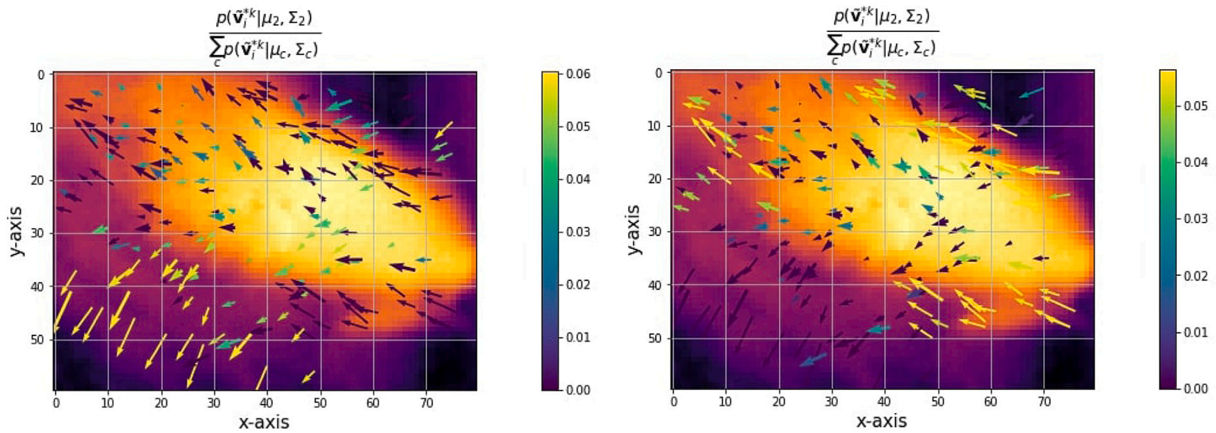
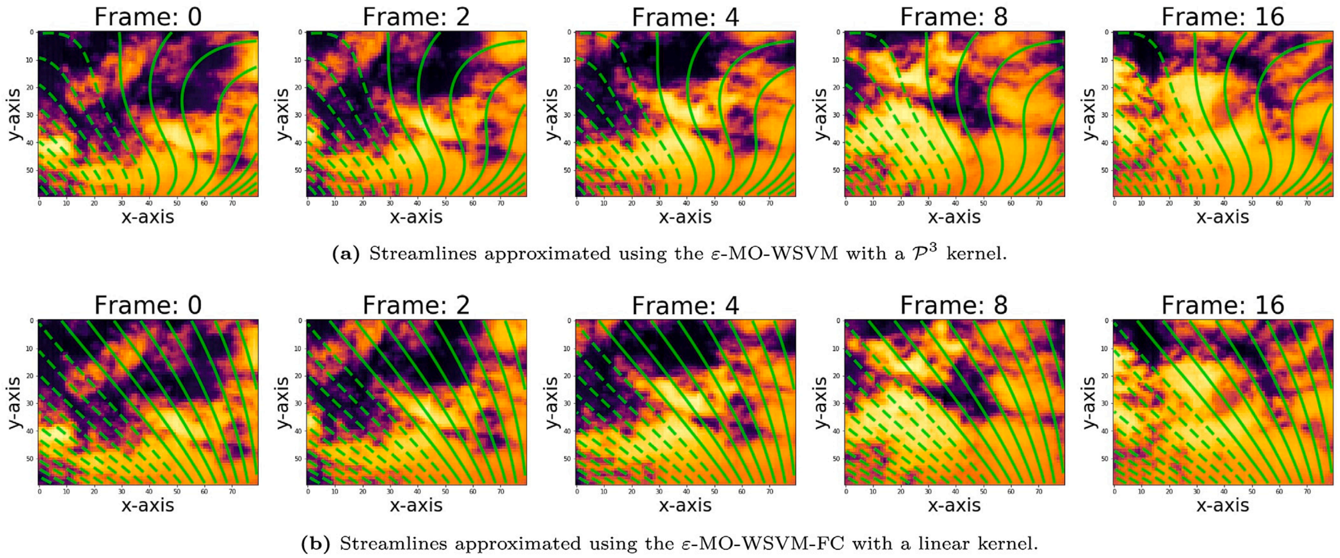


Fig. 4. Velocities selected in Fig. 3 in their corresponding position. The left image vectors show a color intensity corresponding to their posterior probabilities conditional to the upper cloud layer and the right image to the lower cloud layer. The left image shows in yellow the points clearly belonging to the upper layer, while right image shows in yellow the points that are clearly of the lower layer.



(a) Streamlines approximated using the ε-MO-WSVM with a $\mathcal{P}^3$ kernel.

(b) Streamlines approximated using the ε-MO-WSVM-FC with a linear kernel.

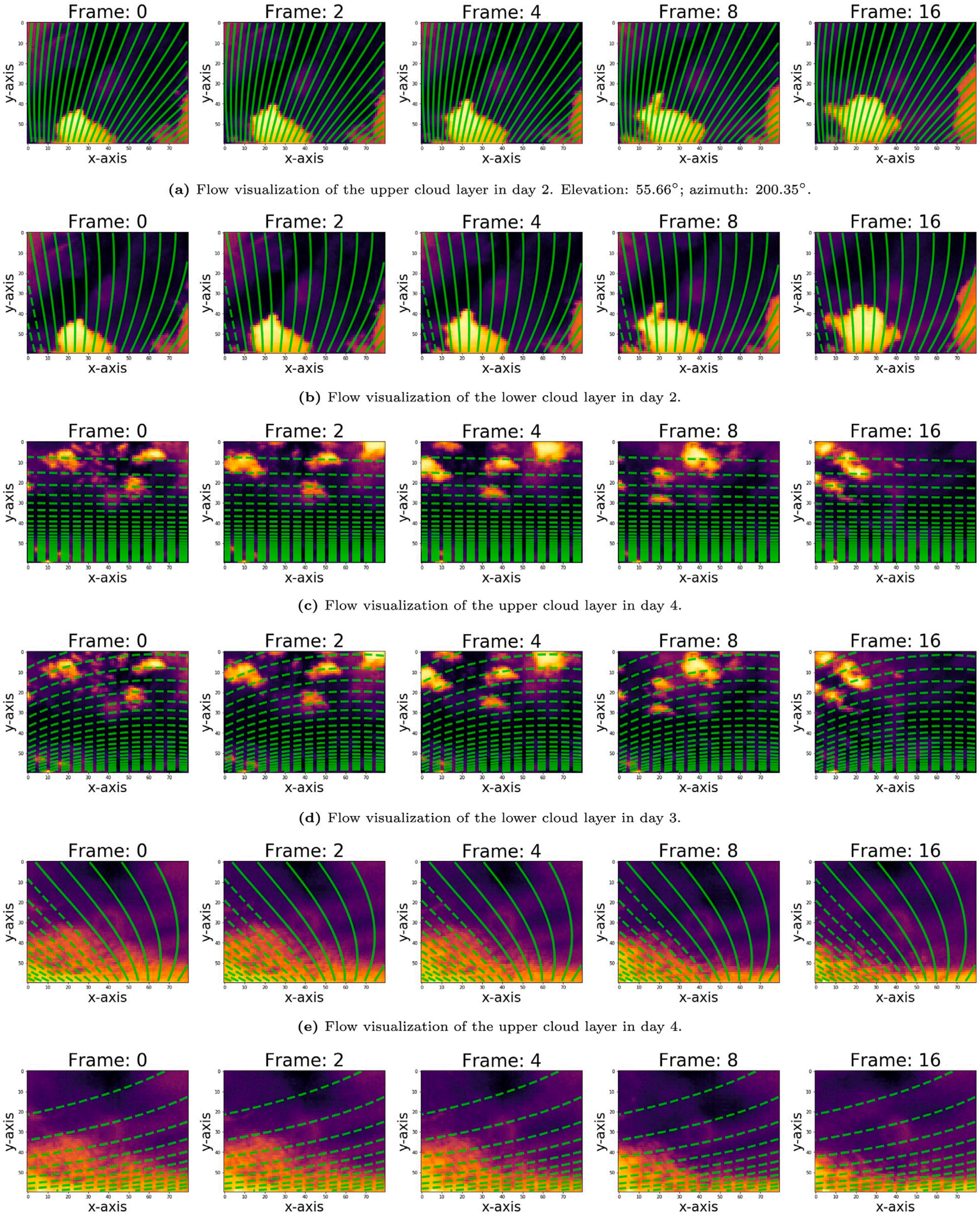
Fig. 5. Comparison between (a) the streamlines computed by a standard Multioutput WSVM (ε-MO-WSVM) and (b) the introduced WSVM with divergence and vorticity constraints (ε-MO-WSVM-FC) in a sequence of images with elevation: 46.74° and azimuth: 174.21°. The images are organized chronologically from the left to the right. The time between frames is 15 s. The distance in the sequences across time is: 0 s, 30 s, 1 min 2 min 4 min. The sequence shows a day when a single cloud layer was detected. The top sequence visualizes a non-realistic approximation of the flow. A compression is induced to the gas in the bottom left of the frame, and an expansion is induced in the top right of the frame.

Table 2
This table shows the optimal sets of parameters obtained cross-validating and training the ε-MO-WSVM-FC in each training image, the results cross-validating the parameters and training the ε-MO-WSVM-FC in each testing image, and the testing results training the ε-MO-WSVM-FC in each testing image using the optimal sets of parameters previously cross-validated in the training data.

ε-MO-WSVM-FC													
	Optimal parameters				Online parameters cross-validation				Fixed optimal parameters				
$\mathcal{K}(\mathbf{x},\mathbf{x}^*)$	C	ε	γ	β	MAE	WMAE	$\nabla\cdot\vec{V}$	$\nabla\times\vec{V}$	MAE	WMAE	$\nabla\cdot\vec{V}$	$\nabla\times\vec{V}$	Time [s]
Linear	38.50	0.19			14.22	**13.36**	**0.0**	**0.0**	14.24	13.35	**0.0**	**0.0**	58.54
RBF	38.52	0.35	13.92		14.55	13.53	30.96	30.86	14.12	**13.05**	136.97	138.80	114.71
$\mathcal{P}^2$	39.72	0.24	3.78	44.8	14.36	13.48	77.22	70.85	14.48	13.59	30.44	30.77	130.92
$\mathcal{P}^3$	12.88	0.22	5.61	8.34	15.34	14.34	$1.74\cdot10^4$	$1.66\cdot10^4$	45.03	44.48	$2.19\cdot10^6$	$1.97\cdot10^6$	145.50

represent the probability of the vector to belong to the corresponding layer. Vectors with a very low probability do not contribute to the solution. Furthermore, the computing time of the ε-WSVM is lower than the ε-MO-WSVM as the Gram matrix dimensions are smaller. The flow divergence and vorticity are negligible when they are approximated using the ε-MO-WSVM-FC, but the computing time is larger. The results are similar between the three models but the ε-MO-WSVM and ε-MO-WSVM-FC models tend to show better performance.

The best result without cross-validation in WMAE is obtained by the ε-MO-WSVM-FC with RBF kernel (see Table 2). The flow approximated by this model has very low vorticity and divergence, which means that the pathlines and streamlines are approximately equivalent. When a trade-off is considered between vorticity, divergence, WMAE, and computing time, the most promising models are the ε-MO-WSVM-FC with linear kernel and RBF kernel. The computing time required for the linear kernel is lower, as the kernel does not have hyperparameters, but



(a) Flow visualization of the upper cloud layer in day 2. Elevation: 55.66°; azimuth: 200.35°.

(b) Flow visualization of the lower cloud layer in day 2.

(c) Flow visualization of the upper cloud layer in day 4.

(d) Flow visualization of the lower cloud layer in day 3.

(e) Flow visualization of the upper cloud layer in day 4.

(f) Flow visualization of the lower cloud layer in day 4.

Fig. 6. Streamlines approximated using the ε-MO-WSVM-FC with a linear kernel. Elevation: 32.15°; azimuth: 180.29°. The sequence of IR images are organized chronologically from the left to the right similar to in Figs. 5(a) and 5(b). The displayed sequences are from days when two different cloud layers were detected. The upper layer of clouds is displayed in the top row of the sequence, the bottom row displays the lower layer.

it is still high for a real-time application. Vorticity and divergence are removed in the approximated flow. On the other hand, the ε-WSVM with linear kernel, which has not flow constraints, is feasible in real-time application but the approximated flow is turbulent. When the pathlines begin to be the same as the streamlines, the flow constraints can be relaxed to reduce the vorticity and divergence within a feasible computing time.

In the implementation of the algorithm, the process of cross-validating the parameters of the ε-MO-WSVM-FC is computationally expensive, and the kernels may have hyperparameters which also require cross-validation. However, the optimal set of parameters is nearly identical during sort sequences. We propose to implement an exhaustive cross-validation in parallel with running the algorithm. This provides a pre-computed set of parameters for the ε-MO-WSVM-FC and the kernel function that can be used in the consecutive images until the online cross-validation is finished.

7. Conclusions

This article introduces a method to visualize wind velocity fields using physical features extracted from infrared images of clouds. The images are recorded using a ground-based infrared camera mounted on a solar tracker that maintains the Sun in the center of the images. The velocity vectors are transformed from the Euclidean frame of reference to the infrared camera non-linear frame of reference. The wind velocity field estimation is based on unsupervised online machine learning methods that independently infer the distribution of the velocity vectors and the height of the clouds. Segmenting and subsampling the velocity vectors provides a computationally tractable solution. The wind velocity field is extrapolated to the entire frame using only information extracted from a cloud. This is achieved with the use of a ε-MO-WSVM which includes flow constraints in the quadratic programming problem formulation.

The methods to compute the motion vectors produce a noisy approximation of the velocity vectors in the frame. It is possible to improve the quality of the velocity vectors adding weights to the least-squares solution in the Lucas–Kanade, and later segmenting the velocity vectors. Once the noise is reduced, a subsample of vectors is sufficient to approximate the velocity field in the entire frame. This makes a real-time implementation of the algorithm for wind flow visualization feasible. This is important, because the wind velocity field visualization predicts the pathlines of the clouds. The extrapolation of the wind velocity field to the entire frame is useful to anticipate where a cloud will be, or where it may appear in the frame. Additional constraints in the SVM yields better results, approximating the wind velocity field in the infrared images.

Further research in this area may focus on predicting the occlusion of the Sun or the attenuation of solar irradiance using the streamline (i.e. pathline) that intercepts the Sun, and the magnitude of the wind velocity field in this streamline. The prediction of the wind velocity field distribution across space and time using Bayesian regression methods is suitable for the selection of the most likely intercepting streamlines. Forecasting solar irradiance is out of the scope of this paper. The prediction methods could use the accumulated distance of the pixels along the streamline (starting from the Sun) divided by the averaged magnitude of the approximated velocity vectors, to estimate the arrival time of the air parcel in that pixel. In the existing literature there are multiple optimization algorithms that might speed up the convergence of the ε-MO-WSVM-FC. These methods may increase the accuracy of very short-term solar irradiance forecasting algorithms that are necessary to optimize the dispatch and storage of energy in power grids that used solar resources.

CRediT authorship contribution statement

Guillermo Terrén-Serrano: Software, Formal analysis, Methodology, Investigation, Data curation, Writing - original draft, Visualization. **Manel Martínez-Ramón:** Writing - review & editing, Conceptualization, Supervision, Project administration, Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been supported by NSF EPSCoR, United States grant number OIA-1757207 and the King Felipe VI endowed Chair, United States. Authors would like to thank the UNM Center for Advanced Research Computing, supported in part by the National Science Foundation, for providing the high performance computing and large-scale storage resources used in this work.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.apenergy.2021.116656.

References

[1] Halu A, Scala A, Khiyami A, González MC. Data-driven modeling of solar-powered urban microgrids. Sci Adv 2016;2(1).

[2] CEC. Clean energy and pollution reduction act - SB 350. 2016, URL https://www.energy.ca.gov/.

[3] ANRE. Feed-in tariff for renewable electricity and solar PV auction. 2017, URL http://www.enecho.meti.go.jp.

[4] NERof South Africa (NERSA). Renewable energy independent power producer programme (REIPPP). 2011, URL http://www.nersa.org.za/.

[5] EC. Renewable energy directive 2018/2001/EU. 2018, URL https://ec.europa.eu.

[6] IEA. Snapshot of global photovoltaic markets. 2015, URL http://www.iea-pvps.org.

[7] Beyea J. The smart electricity grid and scientific research. Science 2010;328(5981):979–80.

[8] Wan C, Zhao J, Song Y, Xu Z, Lin J, Hu Z. Photovoltaic and solar power forecasting for smart grid energy management. CSEE J Power Energy Syst 2015.

[9] Chen X, Du Y, Lim E, Wen H, Yan K, Kirtley J. Power ramp-rates of utility-scale PV systems under passing clouds: Module-level emulation with cloud shadow modeling. Appl Energy 2020;268:114980.

[10] Lappalainen K, Valkealahti S. Output power variation of different PV array configurations during irradiance transitions caused by moving clouds. Appl Energy 2017;190:902–10.

[11] Mateos D, Antón M, Valenzuela A, Cazorla A, Olmo F, Alados-Arboledas L. Efficiency of clouds on shortwave radiation using experimental data. Appl Energy 2014;113:1216–9.

[12] Diagne M, David M, Lauret P, Boland J, Schmutz N. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. Renew Sustain Energy Rev 2013;27(Suppl. C):65–76.

[13] Chou J-S, Truong N-S. Cloud forecasting system for monitoring and alerting of energy use by home appliances. Appl Energy 2019;249:166–77.

[14] Crespi F, Toscani A, Zani P, Sánchez D, Manzolini G. Effect of passing clouds on the dynamic performance of a CSP tower receiver with molten salt heat storage. Appl Energy 2018;229:224–35.

[15] Yang D, Kleissl J, Gueymard CA, Pedro HT, Coimbra CF. History and trends in solar irradiance and PV power forecasting: A preliminary assessment and review using text mining. Sol Energy 2018;168:60–101, Advances in Solar Resource Assessment and Forecasting.

[16] Furlan C, de Oliveira AP, Soares J, Codato G, ao Francisco Escobedo J. The role of clouds in improving the regression model for hourly values of diffuse solar radiation. Appl Energy 2012;92:240–54.

[17] Escrig H, Batlles F, Alonso J, Baena F, Bosch J, Salbidegoitia I, et al. Cloud detection, classification and motion estimation using geostationary satellite imagery for cloud cover forecast. Energy 2013;55(Suppl. C):853–9.

[18] Perez R, Kivalov S, Schlemmer J, Jr. KH, Renné D, Hoff TE. Validation of short and medium term operational solar radiation forecasts in the US. Sol Energy 2010.

[19] Arbizu-Barrena C, Ruiz-Arias JA, Rodríguez-Benítez FJ, Pozo-Vázquez D, Tovar-Pescador J. Short-term solar radiation forecasting by advecting and diffusing MSG cloud index. Sol Energy 2017;155(Suppl. C):1092–103.

[20] Hammer A, Heinemann D, Lorenz E, Lückehe B. Short-term forecasting of solar radiation: a statistical approach using satellite data. Sol Energy 1999.

[21] Ineichen R. Derivation of cloud index from geostationary satellites and application to the production of solar irradiance and daylight illuminance data. Theor Appl Climatol 1999.

[22] Schillings C, Mannstein H, Meyer R. Operational method for deriving high resolution direct normal irradiance from satellite data. Sol Energy 2004.

[23] Jiang H, Lu N, Huang G, Yao L, Qin J, Liu H. Spatial scale effects on retrieval accuracy of surface solar radiation using satellite data. Appl Energy 2020;270:115178.

[24] Prasad AA, Taylor RA, Kay M. Assessment of direct normal irradiance and cloud connections using satellite data over Australia. Appl Energy 2015;143:301–11.

[25] Pedro HT, Coimbra CF. Assessment of forecasting techniques for solar power production with no exogenous inputs. Sol Energy 2012;86(7):2017–28.

[26] Mammoli A, Ellis A, Menicucci A, Willard S, Caudell T, Simmins J. Low-cost solar micro-forecasts for PV smoothing. In: 2013 1st IEEE conference on technologies for sustainability. 2013, p. 238–43.

[27] Voyant C, Notton G, Kalogirou S, Nivet M-L, Paoli C, Motte F, et al. Machine learning methods for solar radiation forecasting: A review. Renew Energy 2017;105(Suppl. C):569–82.

[28] Kalnay E. Atmospheric Modeling, Data Assimilation and Predictability. Cambridge University Press; 2003.

[29] García-Hinde O, Terrén-Serrano G, Hombrados-Herrera M, Gómez-Verdejo V, Jiménez-Fernández S, Casanova-Mateo C, et al. Evaluation of dimensionality reduction methods applied to numerical weather models for solar radiation forecasting. Eng Appl Artif Intell 2018;69:157–67.

[30] Jang H, Bae KY, Park HS, Sung DK. Solar power prediction based on satellite images and support vector machine. IEEE Trans Sustain Energy 2016.

[31] Gohari M, Urquhart B, Yang H, Kurtz B, Nguyen D, Chow C, et al. Comparison of solar power output forecasting performance of the total sky imager and the university of california, san diego sky imager. Energy Procedia 2014;49.

[32] Long C, Slater D, Tooman TP. Total sky imager model 880 status and testing results. Wash, USA: Pacific Northwest National Laboratory Richland; 2001.

[33] Marquez R, Coimbra CF. Intra-hour DNI forecasting based on cloud tracking image analysis. Sol Energy 2013.

[34] Kuhn P, Wilbert S, Prahl C, Schüler D, Haase T, Hirsch T, et al. Shadow camera system for the generation of solar irradiance maps. Sol Energy 2017;157:157–70.

[35] Cheng H-Y. Cloud tracking using clusters of feature points for accurate solar irradiance nowcasting. Renew Energy 2017;104:281–9.

[36] Deneke H, Feijt A, Roebeling R. Estimating surface solar irradiance from METEOSAT SEVIRI-derived cloud properties. Remote Sens Environ 2008;112(6):3131–41.

[37] Mammoli A, Terrén-Serrano G, Menicucci A, Caudell TP, Martínez-Ramón M. An experimental method to merge far-field images from multiple longwave infrared sensors for short-term solar forecasting. Sol Energy 2019;187:254–60.

[38] Wizelius T. Developing wind power projects: theory and practice. Earthscan; 2007.

[39] Glickman TS, Zenk W. Glossary of meteorology. AMS (American Meteorological Society); 2000.

[40] Randel WJ, Wu F, Gaffen DJ. Interannual variability of the tropical tropopause derived from radiosonde data and NCEP reanalyses. J Geophys Res: Atmos 2000;105(D12):15509–23.

[41] Shaw JA, Nugent PW, Pust NJ, Thurairajah B, Mizutani K. Radiometric cloud imaging with an uncooled microbolometer thermal infrared camera. Opt Express 2005;13(15):5807–17.

[42] Shaw JA, Nugent PW. Physics principles in radiometric infrared imaging of clouds in the atmosphere. Eur J Phys 2013;34(6):S111–21.

[43] Nugent PW, Shaw JA, Pust NJ. Correcting for focal-plane-array temperature dependence in microbolometer infrared cameras lacking thermal stabilization. Opt Eng 2013;52(6). 1 – 8 – 8.

[44] Thurairajah B, Shaw JA. Cloud statistics measured with the infrared cloud imager (ICI). IEEE Trans Geosci Remote Sens 2005;43(9):2000–7.

[45] Nugent PW, Shaw JA, Piazzolla S. Infrared cloud imaging in support of Earth-space optical communication. Opt Express 2009;17(10):7862–72.

[46] Chow CW, Urquhart B, Lave M, Dominguez A, Kleissl J, Shields J, et al. Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. Sol Energy 2011.

[47] Fu C-L, Cheng H-Y. Predicting solar irradiance with all-sky image features via regression. Sol Energy 2013;97:537–50.

[48] Kong W, Jia Y, Dong ZY, Meng K, Chai S. Hybrid approaches based on deep whole-sky-image learning to photovoltaic generation forecasting. Appl Energy 2020;280:115875.

[49] Deng C, Li Z, Wang W, Wang S, Tang L, Bovik AC. Cloud detection in satellite images based on natural scene statistics and gabor features. IEEE Geosci Remote Sens Lett 2019;16(4):608–12.

[50] Terrén-Serrano G, Martínez-Ramón M. Data acquisition and image processing for solar irradiance forecast. 2020, URL arXiv:2011.12401.

[51] Reda I, Andreas A. Solar position algorithm for solar radiation applications. Sol Energy 2004;76(5):577–89.

[52] Vapnik VN. Statistical learning theory. Wiley-Interscience; 1998.

[53] Muralikrishna IV, Manickam V. Chapter fourteen - air pollution control technologies. In: Environmental management. Butterworth-Heinemann; 2017, p. 337–97.

[54] Murphy KP. Machine learning: A probabilistic perspective. The MIT Press; 2012.

[55] Bishop CM. Pattern recognition and machine learning. Springer; 2006.

[56] Nocedal J, Wright SJ. Numerical optimization. 2nd ed.. New York, NY, USA: Springer; 2006.

[57] Terrén-Serrano G, Martínez-Ramón M. Comparative analysis of methods for cloud segmentation in infrared images. 2020, URL arXiv:2012.06930.

[58] Lucas BD, Kanade T. An iterative image registration technique with an application to stereo vision. In: IJCAI. 1981.

[59] Horn BK, Schunck BG. Determining optical flow. Artif Intell 1981;17(1–3):185–203.

[60] Farnebäck G. Two-frame motion estimation based on polynomial expansion. Image Anal 2003;363–70.

[61] Adrian L, Adrian R, Westerweel J. Particle image velocimetry. Cambridge aerospace series, Cambridge University Press; 2011.

[62] Baker S, Gross R, Ishikawa T, Matthews I. Lucas-kanade 20 years on: A unifying framework: Part 2. Int J Comput Vis 2003;56:221–55.

[63] Jensen JLWV. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. Acta Math 1906;30:175–93.

[64] Besag J. On the statistical analysis of dirty pictures. J R Stat Soc B 1986;48(3):48–259.

[65] Drucker H, Burges CJC, Kaufman L, Smola AJ, Vapnik V. Support vector regression machines. In: Mozer MC, Jordan MI, Petsche T, editors. Advances in neural information processing systems 9. MIT Press; 1997, p. 155–61.

[66] Schölkopf B, Smola AJ, Williamson RC, Bartlett PL. New support vector algorithms. Neural Comput 2000;12(5):1207–45.

[67] Elattar EE, Goulermas J, Wu QH. Electric load forecasting based on locally weighted support vector regression. Trans Syst Man Cyber C 2010;40(4):438–47.

[68] Cortes C, Vapnik V. Support-vector networks. In: Machine learning. 1995, p. 273–97.

[69] Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis. New York, NY, USA: Cambridge University Press; 2004.

[70] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput 2004;14(3):199–222.

[71] Han X, Clemmensen L. On weighted support vector regression. Qual Reliab Eng 2014.

[72] Lamb D, Verlinde J. Physics and chemistry of clouds. Cambridge University Press; 2011.

[73] Granger R. Fluid mechanics. Dover books on physics, Dover Publications; 1995.

[74] Dragomir SS, Cerone P, Sofo A. Some remarks on the trapezoid rule in numerical integration. RGMIA Res Rep Collect 1999;2(5).

[75] Terrén-Serrano G, Bashir A, Estrada T, Martínez-Ramón M. Girasol, a sky imaging and global solar irradiance dataset, dryad, dataset. 2021, URL https://doi.org/10.5061/dryad.zcrjdfn9m.

[76] Rasmussen CE, Williams CKI. Gaussian processes for machine learning (adaptive computation and machine learning). The MIT Press; 2005.

[77] Tikhonov AN, Arsenin VY. Solutions of ill-posed problems. Washington, D.C.: John Wiley & Sons, New York: V. H. Winston & Sons; 1977, p. xiii+258, Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

[78] Bonilla EV, Chai K, Williams C. Multi-task Gaussian process prediction. In: Platt J, Koller D, Singer Y, Roweis S, editors. Advances in neural information processing systems, vol. 20. Curran Associates, Inc.; 2008, p. 153–60.