

Development of a low-cost, short-term solar irradiance forecasting system

Miguel Gomes Lopes
miguel.lopes@ist.utl.pt

Instituto Superior Técnico, Lisbon, Portugal

May 20, 2015

Abstract

An innovative, low-cost, ground based sky imager prototype was fully developed at Instituto Superior Técnico, in Lisbon, along with all the software needed to control it. Based on this sky imager, a method for intra-hour, sub-kilometer irradiance forecasting is presented, an important component of solar energy applications. Sky images are taken every 20s and autonomously processed to remove lens distortion. Cloud pixels are then identified, by means of the normalized red-blue-ratio and the LAB color space. Using two consecutive images, the velocity vector for the clouds is obtained through the application of the cross-correlation method. By merging the information from the cloud positions and the velocity vectors, we are able to predict the direct normal irradiance for up to 6 minutes ahead. The solar forecasts derived are analyzed and validated in terms of root mean square error deviations in relation to measured values and compared to the performance of a persistence model. Our results show a significant improvement over the persistence models for time horizons greater than 1 minute. For less than 1 minute horizons, the uncertainty of cloud identification in the circumsolar region makes it very difficult to beat persistence models.

Keywords: irradiance forecasting, sky images, cloud identification, particle image velocimetry, camera calibration

1. Introduction

The two main challenges to high penetration rates of photovoltaic (PV) systems into the electric grid are variability and uncertainty, i.e., the fact that PV output exhibits variability at all timescales (from seconds to years) and the fact that this variability itself is difficult to predict [1]. Of particular interest to the energy industry are sudden and widespread changes in irradiance, typically caused by the presence of clouds. Reduction in the uncertainty of solar generation through accurate short-term solar forecasting greatly reduces solar integration costs [2]. To date, short-term accurate solar forecasts have been evasive and the problem proven to be complex due to the impact of weather patterns on the intensity of solar irradiance; most notably local cloud fields.

Different solar irradiance forecast methodologies have been proposed for various time horizons. From 6h up to several days ahead, solar irradiance forecasts rely on Numerical Weather Prediction (NWP) models [3, 4, 5, 6]. In the 1-4h forecast range, Satellite imagery is the best forecasting technique available [4, 7, 8].

In order to achieve high temporal and spatial resolution for intra-hour forecasts, NWP and satellite forecasts are currently inadequate [9]. Ground observations using a sky imager (SI) present an opportunity to fill this forecasting gap and deliver a sub-kilometer view of cloud shadows over a central PV power plant or an urban distribution feeder. This method can be used to forecast from real time (nowcast) up to 10 to 30 minutes ahead by applying image processing and

cloud tracking techniques to sky photographs. Typical components of a SI are a charge coupled device (CCD) camera, a fish-eye lens or a spherical mirror, an environmental housing, and possibly a solar occulter depending on the choice of CCD sensor and application.

Solar forecasting based on SI analysis generally consists of four steps: acquisition of a sky image in the vicinity of the forecast site; analysis of sky image data to identify clouds; estimation of cloud motion vectors using successive images; use of cloud location and motion vector data for short term deterministic or probabilistic cloud cover, irradiance, and power forecasting.

Cloud detection using SIs is generally based on a thresholding technique that utilizes the camera's red-green-blue (RGB) channel magnitudes to determine the red-blue ratio (RBR) (Shields algorithm [10]). Increased molecular (Rayleigh) scattering of shorter wavelengths causes the clear sky to be blue and the signal measured in the blue channel to be larger. Since clouds scatter the visible wavelengths more evenly, the red signal is similar to the blue signal. The Shields algorithm uses fixed ratio thresholds to identify opaque clouds; thin clouds are detected through a comparison with a clear sky background RBR library as a function of solar angle, look angle and site location. However, other methods have been applied to identify clouds. Souza-Echer et al. [11] used saturation in the hue, saturation and luminance (HSL) color space with fixed thresholds for cloud detection. Cazorla et al. [12] classified clouds based on neural networks. Finally,

Li et al. [13] developed a hybrid thresholding technique (HYTA) that is based on both fixed and adaptive thresholding techniques for cloud detection. In 2012, Ghonima et. al [14] performed a relevant method that distinguished between not only thin and thick clouds, but also aerosols, adding a haze correction factor to the classification model to account for variations in the level of aerosol.

One of the most relevant recent studies of solar forecasting using SI is [9], where irradiance nowcasting was proposed to achieve a high temporal and spatial resolution through the analysis of sky cover images. Their cloud detection technique is based on the concept of RBR, previously explained. Instead of using just a single RBR threshold for the entire image, a clear sky library (CSL based on [15]) as a function of zenith and sun-pixel-angle was calculated from images on a cloud-free day. A clear sky background image is then generated for each sky image based on the current solar zenith angle. Cloud velocity and direction of motion is determined through the cross-correlation method (CCM) applied to two consecutive sky images [16].

Our goal is to develop a low cost sky imager to predict DNI for intra-hour and sub-kilometer resolutions. We do not aim to compete with expensive, commercially available SI's, but rather prove that it is possible to acquire useful information to the solar energetic industry without a big financial investment.

Section 2 addresses all aspects related to the conception SI and Section 3 describes all the algorithms developed in order to control the prototype and analyze the sky images. Section 4 show the results obtained for each step of the method and for the irradiance, by calculating the root mean square errors (RMSE) in comparison to the measured irradiance and by comparing our method with a persistence model. Conclusions follow in section 5.

2. Hardware

The biggest problem with commercially available SI's is their price, in the region of tens of thousands of dollars. Therefore, we decided to build our own low-cost SI.

A structure like the TSI found in [9] was discarded, since curved mirrors with high reflectivity, like the ones used, are very expensive. A geometry like the one found in [17], although it doesn't have any moving parts, the solar obstructor needs to cover all the sun's positions during the day, which results in a very large region for which the information is lost. Furthermore, since the camera is looking up, a quality fish-eye lens is needed, which is very expensive. The third type of structure found in the literature [12] needs to move the solar obstructor with two degrees of freedom, which exponentially increases the complexity. A structure without a solar obstructor, like the one found in [18], was also discarded, since this would lead to the degradation of the camera and to undesirable effects in the photographs.

We therefore came up with an original prototype

that allows to obstruct the sun while maintaining the simplicity and low cost of the SI, shown in Figure 1.



Figure 1: Prototype placed where the photographs were retrieved.

Figure 2 shows the mechanical interface of the prototype.

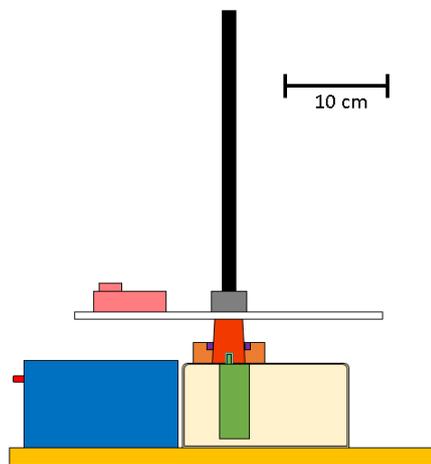


Figure 2: Mechanical interface, longitudinal view.

A vertical pole (black) is used as the sun obstructor and placed in the center of a rotating platform (white), on which the camera (pink) is placed, in the shadow of the pole. The rotation of the platform has the speed of the sun, allowing to keep the camera in the shadow while retrieving images. The box filled in beige protects the motor (green) and all the electronics. A cone (red) is inserted inside a bearing (purple) to transfer the rotation from the axis of the motor to the platform. The voltage source shown in blue powers all the electronics. The cost of the whole system's materials lies around 300€. The SI is very portable (3kg).

The **camera** used to retrieve the sky photographs was a *GoPro Hero3 White Edition* [19]. The camera lens is ultra-wide, with a field of view (FOV) of 150° diagonally (with a 4:3 format). It should be enough to observe the useful regions of sky, since for zenith angles greater than 65° the Mie dispersion whitens the horizon and doesn't allow for cloud identification [9].

The **motor** used to rotate the platform was the model EMG30 from Robot Electronics [20]. It has an

included gearbox of 1:30, as well as an encoder with two Hall effect sensors (one for each direction of motion) with a theoretical precision of 1° . To communicate between the computer and the motor we resort to the serial port by using the MD25 board from the same manufacturer.

In order to retrieve **irradiance measurements**, we used 6 PV panels belonging to the PEC project [21], in IST Taguspark. The energy generated by these panels is stored in a battery bank, therefore, we can assume that the current into the battery represents the variations in the irradiance, although this system may not accurately represent DNI, since there are other types of irradiance that influence PV panels.

3. Software

Figure 3 shows the flowchart of all the software needed to control the hardware and analyze the photographs in order to obtain the solar irradiance.

Since the Sun rotates 0.25° every minute and we only have 1° precision, the **rotation of the motor** must happen every 4 minutes. In order to rotate such a small angle, we first get the value of the encoder and set the motor speed to its minimum value. After waiting a very small time interval, we read the encoder again. If we obtain a difference between the initial and final encoders of 1, then the motor rotated 1° . If we obtain a value of zero, then the motor didn't rotate 1 degree yet and the cycle is repeated. If the difference is over 1, then the motor rotated too much and we need to repeat the process, this time with a negative velocity.

After rotating the motor, a **photo must be obtained** through wireless, since the camera doesn't allow for USB transfer while turned on.

In order to analyze the photographs, we first decide if the photo needs to be rotated, since we need the observed region of the sky to be the same for two consecutive photographs. Therefore, if the motor rotated in the present iteration, then the **photograph is rotated** by 1° by using a two dimensional rotation matrix and nearest-neighbor interpolation to obtain the colors.

As explained, the ultra-wide lens of the camera introduces severe distortion to the photographs. In order to undistort them into a rectilinear coordinate system, a **camera calibration** needed to be performed. The goal with this calibration was to determine the intrinsic properties of the camera and its orientation in relation to the world coordinate system. The algorithm was developed by Bouguet [22] as a derivation from the studies of Zhang [23] and Heikkil [24]. In order to apply this method we only need a set of images in which the camera is looking into a checkerboard pattern, with different orientations for each photograph. The origin of the pattern is then detected and, for each vertex of the pattern, the coordinates u, v (in pixels) and (X, Y, Z) (in world units) is generated (we need to know the length of the side of the squares).

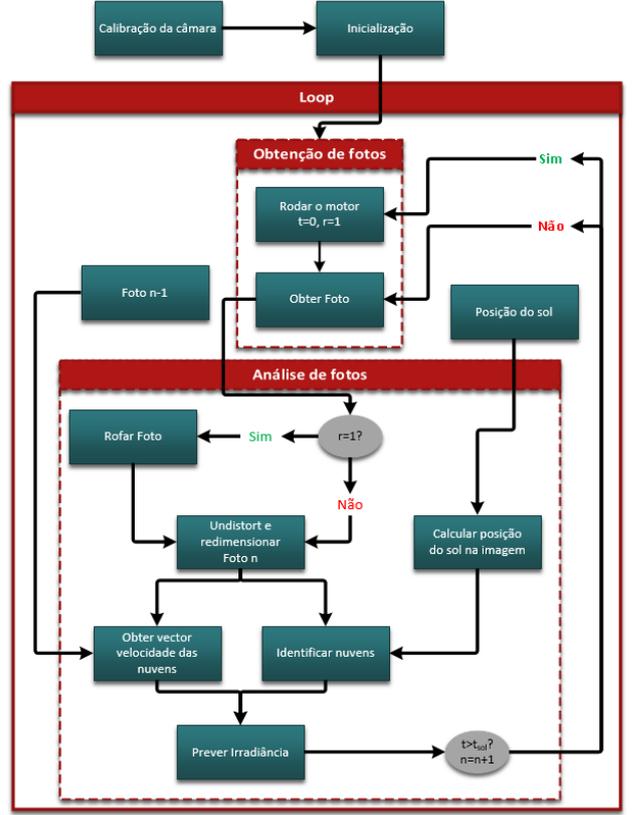


Figure 3: Flowchart of all the software developed for this work.

We then assume a pinhole model [23]:

$$w \begin{bmatrix} u & v & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} R \\ t \end{bmatrix} K, \quad (1)$$

where w is an arbitrary scale factor. The extrinsic parameters are the translation vector of the camera in relation to the world system units t and the rotation vector R . The latter can be converted into a rotation matrix 3×3 using Rodrigues formula [25].

The intrinsic matrix K is defined as:

$$\begin{bmatrix} f_u & 0 & 0 \\ s & f_v & 0 \\ c_u & c_v & 1 \end{bmatrix}, \quad (2)$$

where f_u and f_v are the focal distance, in pixels, vertically and horizontally, c_u and c_v are the coordinates of the optical center, in pixels, and s is the skew parameter.

The algorithm then estimates the intrinsic and extrinsic parameters assuming that there is no distortion. Using the obtained values as the initial assumption, the algorithm estimates the parameters again by the least-square method, but this time assuming radial distortion [24]. The radial distortion coefficients k_1 and k_2 are defined as:

$$u = u_{ud}(1 + k_1 \times r^2 + k_2 \times r^4) \quad (3)$$

$$v = v_{ud}(1 + k_1 \times r^2 + k_2 \times r^4), \quad (4)$$

where u and v are the coordinates of the distorted pixel, u_{ud} and v_{ud} the coordinates of the undistorted pixel, and $r^2 = u^2 + v^2$. It should be noted that the intrinsic properties of the camera are valid for every photo taken with the same camera (with the same settings). On the contrary, the extrinsic parameters are dependent on the location and orientation of the camera.

In order to obtain the **undistorted photo**, we only need to solve equation 3 for each pixel, from which we obtain u_{ud} and v_{ud} .

With the photograph undistorted, we then need to **shrink the size of the image**, since the analysis of the photographs would take too long with the original size. We use a bicubic interpolation to reduce the image to 1/3 of its original size.

In order to **obtain the solar position** (zenith and azimuth) of the sun as a function of the time the photograph was taken, we run the algorithm developed by Reda and Andreas [26]. This information is then used to identify which pixel of the image corresponds to the Sun. In order to do that, we consider the geometry of Figure 4.

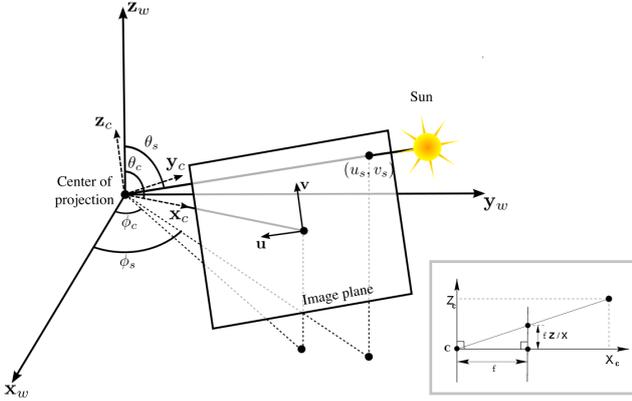


Figure 4: Geometry considered to identify the sun in the image.

By similarity of triangles, we find that:

$$u_s = \frac{-f_u \times y'_s}{x'_s} \quad (5)$$

$$v_s = \frac{-f_v \times z'_s}{x'_s}, \quad (6)$$

where $s' = (x'_s, y'_s, z'_s)$ are the 3D coordinates of the Sun in the camera's frame of reference.

The sun coordinates (x_s, y_s, z_s) as a function of (θ_s, ϕ_s) are:

$$s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} \sin\theta_s \cos\phi_s \\ \sin\theta_s \sin\phi_s \\ \cos\theta_s \end{bmatrix}. \quad (7)$$

s' results from the rotation of s by θ_c and ϕ_c :

$$s' = R_2 R_1 s, \quad (8)$$

where R_1 is the typical rotation matrix around zz by ϕ_c degrees counterclockwise and R_2 is the rotation matrix around yy by θ_c degrees clockwise.

In order to calculate the **velocity vector**, an essential intermediate step for the irradiance forecasting, we used the MATLAB toolbox MPIV (MATLAB Particle Image Velocimetry) developed by Mori and Chang [27]. The algorithm computes the velocity vectors for a pair of sequential images by applying the 2D normalized cross-correlation method (CCM):

$$C(\Delta X, \Delta Y) = \frac{\sum_{i=1}^N \sum_{j=1}^N [f_1(X_i, Y_j) - \bar{f}_1] [f_2(X_i + \Delta X, Y_j + \Delta Y) - \bar{f}_2]}{\sqrt{\sum_{i=1}^N \sum_{j=1}^N [f_1(X_i, Y_j) - \bar{f}_1]^2} \sqrt{\sum_{i=1}^N \sum_{j=1}^N [f_2(X_i + \Delta X, Y_j + \Delta Y) - \bar{f}_2]^2}}$$

, where f_1 and f_2 are the windows of size N of each image on which to perform the CCM and the overbar represents the mean value. In order to compute the peak of the correlated results (C), a Gaussian fit is performed. In order to remove stray vectors, we filter the velocity field vectors using the median and the median standard deviation. A representative velocity vector is then computed by averaging the velocity field.

The last step before calculating the irradiance is to **identify the clouds** in the images. We developed our own algorithm based on three different color spaces: RGB, HSL and LAB. Before identifying the clouds, we first need to calculate the mask that will remove every pixel that doesn't correspond to cloud or sky. For this we use three different thresholds corresponding to the channels H, S and B. These thresholds are not fixed, rather they are a function of the median of the channel for the whole image. In order to identify the clouds we use the NRBR:

$$NRBR = \frac{R - B}{R + B}. \quad (9)$$

We divided the images in 4 different rings centered on the sun's position and, for each ring, we use a fixed NRBR threshold. Despite the fact that this ratio correctly identifies thick clouds, it also identifies thin clouds as clouds, which we don't want, since thin clouds only decrease DNI by roughly 20%. Therefore, in order to distinguish between thin and thick clouds, we resorted to the LAB color space [28], specially designed to describe the difference between colors as perceived by the human eye. The euclidean distance between 2 colors in this space (referred to as the Delta-E) is higher for when the colors are more distinct between each other, which doesn't happen in any other color space. Hence, we calculate the Delta-E for every pixel previously identified as cloud, comparing it with the average color of the cloud region. For thin clouds (more bluish than the rest of the clouds), the Delta-E value is higher. We can then use a thresholding technique to eliminate thin clouds from the cloud identification. Finally, all non-identified pixels (corresponding to the mask) are interpolated by a least-squares method.

The last step is to **predict DNI**. There are two main factors influencing DNI: optical depth of the clouds and the cloud cover in a circle of around 5° from the sun's position. The optical depth was already accounted for when identifying cloud pixels. Figure 5 helps to un-

derstand visually our method for predicting solar irradiance. A "corridor" (shadow region in the figure) is constructed from the sun in the direction contrary to the velocity vector (shown in green). If we multiply this vector by the length of time we want to look ahead, we get a displacement vector that, if applied on the sun, gives us the center of a region which should have a radius of 5° . The cloud cover in this region is simply the ratio between the amount of cloud pixels by the total amount of valid pixels. The relationship between this quantity and DNI is linear:

$$DNI_{pred} = -DNI_{max}R_c + DNI_{max}, \quad (10)$$

where DNI_{max} is the irradiance in clear sky conditions. We do not compute a clear sky irradiance model since our measured values for comparison are in amperes. Therefore, for each photograph, we calculate the maximum value of the measured results in the last 15 minutes and use this value for DNI_{max} . Note that the length of the corridor gives us the maximum temporal forecasting horizon.

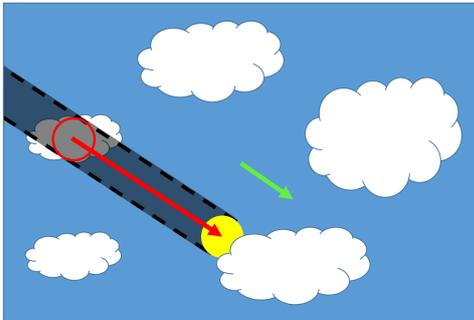


Figure 5: Construction used to predict irradiance.

4. Results

We used two different and independent sets of data. For the first one, the training set for which the algorithms were developed and improved on, we took 3564 photographs between 19/02/2015 and 18/03/2015, for a total of 11 different days. We only considered photographs valid if their total cloud cover was greater than 10% and smaller than 90%. The validation data set was composed of 1650 valid images taken in 19/03/2015. Our SI was placed on the top of the PEC container, near the PV panels. The elevation of the camera was set to 22° . The geographical coordinates are 38.737316° N and 9.303816° W, with an altitude of 168 metros.

The photographs taken are saved as JPEG with a resolution of 2592×1944 and a bit depth of 24 bits per pixel. After removing distortion using the algorithm explained in section 3, we obtain an image with a resolution of 2392×3548 pixels (see Figure 6) After resizing the image, it becomes a 790×1171 image.

As explained in section 3, the undistortion of the image leads to loss of information. The final FOV of the undistorted image was found to be 75.30° vertically, 88.90° horizontally and 100.50° diagonally.



(a) Original image with a resolution of 2592×1944 .



(b) Image after undistortion. The green area shows the size of the original image and the red area the size of the final undistorted image, with a resolution of 2392×3548 pixels.

Figure 6: Undistortion of a calibration image used.

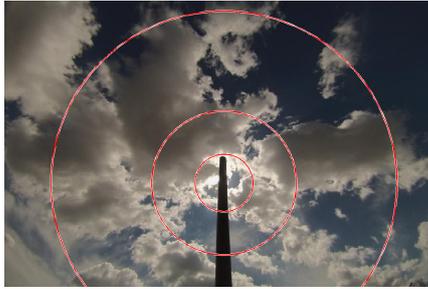
4.1. Cloud decision algorithm

Figure 7 shows the result of the applying the cloud decision algorithm to the original photograph shown in Figure 7(a). We can see the image segmentation around the Sun as well as the NRBR and the Delta-E values. As we can see, the NRBR (Figure 7(b)) identifies thin and thick clouds ($NRBR > -0.05$). To distinguish between these two type of clouds, we proceed to calculate Delta-E for the region identified as clouds, resulting in Figure 7(c). The bluer a pixel is, the closest its color is from the average cloud color. We can then assume that the red structures are different from the blue structures.

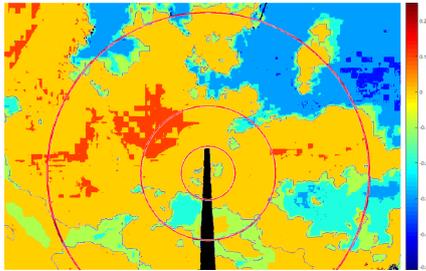
If we subtract these red Delta-E regions from the NRBR cloud decision, we obtain the final cloud decision, shown in Figure 7(d).

In general we found that all the thick clouds of the second, third and fourth rings are correctly identified. In the circumsolar region, however, the identification of the clouds is only correct when the Sun is covered by clouds, since otherwise the blooming effect saturates the pixels and doesn't allow for cloud identification. Sometimes, even when the sun was covered by thick

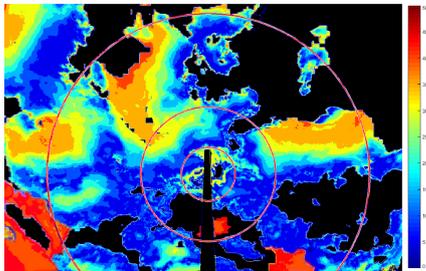
clouds, reflections in very white clouds caused the failure of the algorithm. The identification of thin and diffuse clouds were generally more erroneous. We found that if the amount of thin clouds in the FOV is smaller than thick clouds, the algorithm performs well; if the contrary is true, the algorithm fails, since the mean color of the cloud region will be too close to the thin clouds to distinguish them by means of Delta-E. The pixels that were not identified (belonging to the masks) were then interpolated.



(a) Original photograph.



(b) NRBR. Hot colors represent clouds.



(c) Delta-E. Hot colors represent biggest difference in color.



(d) Final cloud decision.

Figure 7: Application of the algorithm for cloud identification.

The algorithm to identify masks was found to perform very well for all scenarios. The ratio mask/total pixels of all the valid photographs was found to be less than 1%.

4.2. Velocity vectors calculation

The MPIV was applied to two consecutive images, as explained in section 3 (Figure 8). We used the undistorted and resized photographs in the red channel, since it's in this channel that the contrast between sky and clouds is greater.

The reason for not applying the algorithm directly in the cloud identification decision is due to the fact that this would incur in error propagation. If we call the image pair separated by 20s f_1 and f_2 , then the image shown is f_2 .

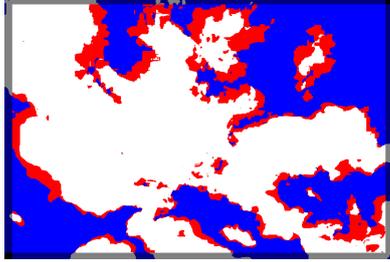


Figure 8: Application of the algorithm for determining velocity vectors. The Sun is shown in yellow and the mean velocity vector in green.

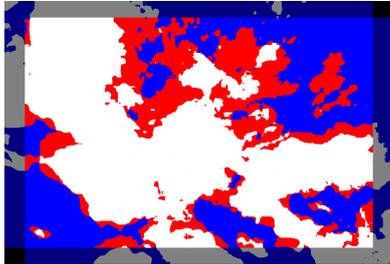
As we can see, in the regions where there is only sky the associated errors of the vectors are greater and these vectors are excluded. The same happens for the obstructor. We can also see in the bottom left corner that vectors are also non-valid, which means that the *cirrus* cloud in this region is not moving, contrary to the rest of the clouds. Therefore, we can see that we made a good decision to don't identify these thin clouds as clouds, since the representative vector is not valid.

To evaluate the quality of the results we decided to use the cloud decisions, since it's impossible to evaluate the original image quantitatively. We decided to superimpose f_1 and f_2 , moving f_1 accordingly to the velocity vector v_v obtained. This comparison is shown in Figure 9(a), where the red regions represent the errors. Figures (c) and (b) in 9 show the comparison between f_2 and f_3 , where f_3 is the image corresponding to the time horizon Δt . Assuming the same velocity field obtained from f_1 and f_2 , f_2 was moved according to $v_v \times \Delta t$. Analyzing the sequence of images, we can see that the errors are not caused by lag between the clouds (as would happen if the errors were due to the velocity vectors), but rather due to the modifications in cloud shape over time. Of course, cloud identification

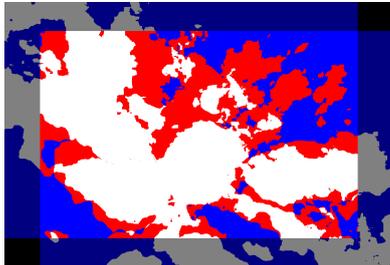
errors may also add to this error. We can clearly see from Figure 9(a) to 9(c) that the errors in the bottom left corner of the image do not increase significantly. The problem lies in the central/superior region of the image, where the clouds (in white) disappear over time. The original shape can be seen in the background, in red.



(a) Comparison between f_1 and f_2 , with f_1 moved according to $v_v \times \Delta t$.



(b) $\Delta t = 60s$. Comparison between f_2 and f_3 , with f_2 moved according to $v_v \times \Delta t$.



(c) $\Delta t = 100s$. Comparison between f_2 and f_3 , with f_2 moved according to $v_v \times \Delta t$.

Figure 9: Visual validation of the velocity vectors for different time horizons.

Figure 10 shows the evolution of each component of the velocity vector along the day. Despite the fluctuations, we can see that the velocities lie in an approximated interval of 3pixels/s. Large variations of the size of the velocity vector are a red flag for potential problems, since this is a very unnatural behavior.

4.3. Irradiance forecasting

Figure ?? shows the cloud decision for an image f_2 obtained at t_0 and the v_v (with f_1 and f_2 20 seconds apart) in green. Each circle represents the circumsolar region relevant to DNI forecasting that results from the application of $-v_v \times \Delta t$ in the solar position. Each

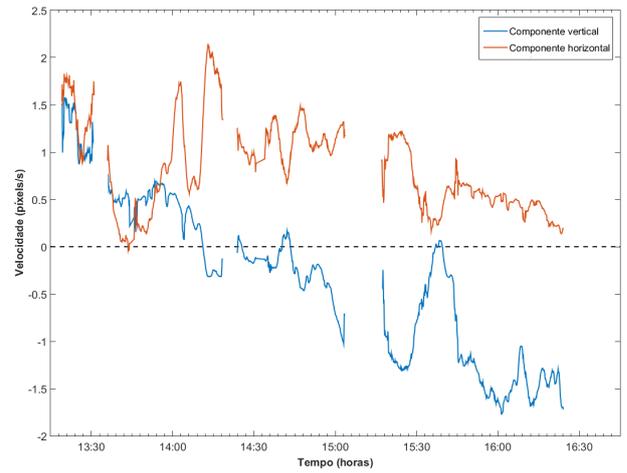


Figure 10: Evolution of each component of the velocity vector along the day.

circle corresponds to a different Δt (1 to 6 minutes). As we can see, the only relevant region of the sky to the prediction is the corridor created by the circles. If we wanted to increase the maximum available time horizon we would need to, when initializing the process for obtaining photographs, align the diagonal of the image with the direction of cloud motion.

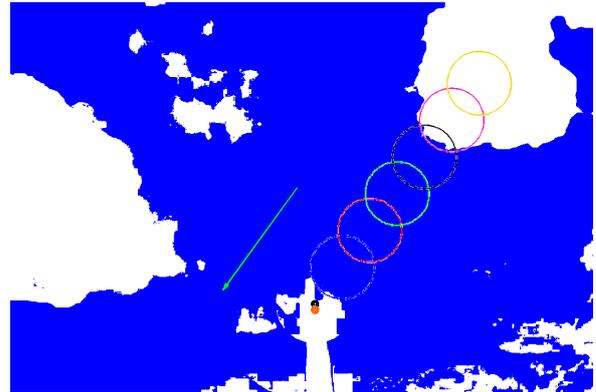


Figure 11: Construction to calculate irradiance. The regions of interest for the different time horizons (1 to 6 minutes) are shown in circles. The black dot shows the solar position for t_0 and the orange one for $t_0 + \Delta t$.

In order to visually evaluate the performance of this method, we compared the regions inside the circles with the circumsolar region in the real photographs obtained to $t_0 + \Delta t$. We expect that the cloud cover is the same for both. We found that for $\Delta t = 1min$, the circle contains 28% of interpolated pixels, which usually leads to large forecasting errors. For $\Delta t = 2min$, $3min$ and $4min$ (red, green and black circles), the forecasting appears to be correct. For $2min$ and $3min$ we found a cloud ratio of 0, meaning clear sky, while the $4min$ horizon shows a ratio of 0.34.

We then proceeded to the calculation of DNI for the photographs taken 19/03/2015 for different time hori-

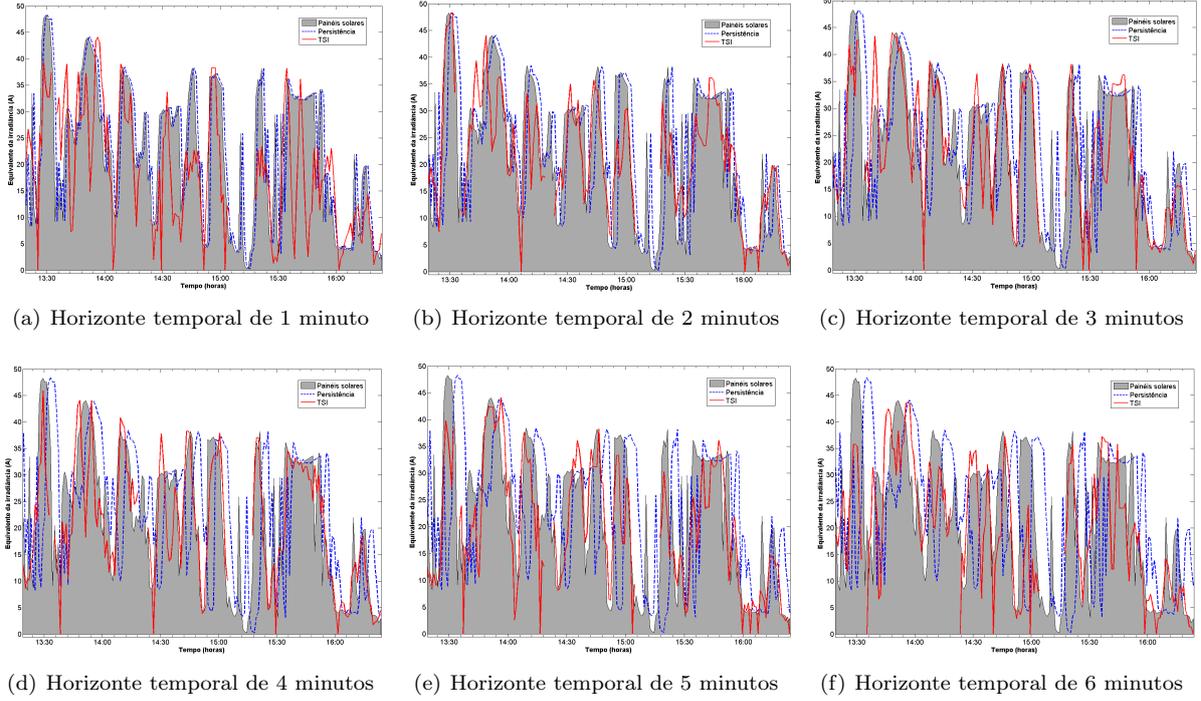


Figure 12: Predicted DNI calculated from the images taken in 19/03/2015 for different temporal horizons.

zons, shown in Figure 12. We also show the values measured with the PV panels, as well as the persistence model forecast.

In order to evaluate the performance of the method we calculate the root mean square error (RMSE) for each of the temporal series. Figure 13 shows the obtained values for RMSE for each of the time horizons, both for our method as well as for the persistence model. We can see significant improvements of our method for all time horizons, with the exception of 1 minute, for which the uncertainty caused by the obstructor rises the errors. The RMSE for our method is minimum for a 3 minutes time horizon, the optimal forecasting horizon.

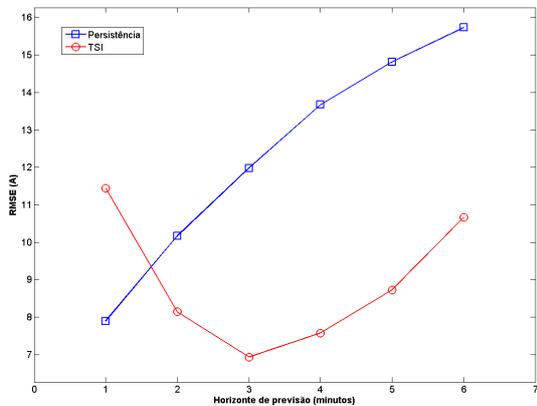


Figure 13: Comparison of RMSE errors of our method and the persistence model.

4.4. Main error sources

Figure 14 shows the error $e_i = I_{i,pred} - I_{i,obs}$, where $I_{i,pred}$ and $I_{i,obs}$ are the i^{th} term of the predicted and observed irradiance. The following discussion is based, mostly, on the visual analysis of the different photographs for when e_i assumes large values. The analysis was performed for the 6 temporal horizons considered.

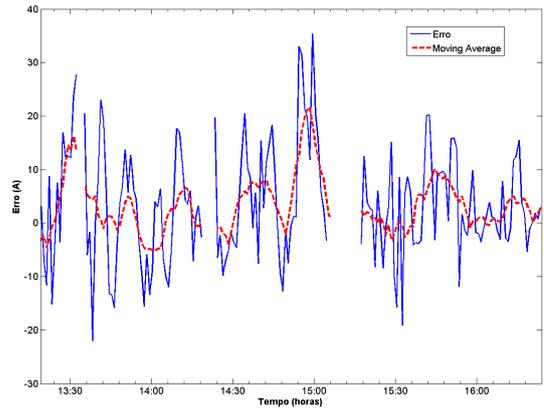


Figure 14: Errors e_i for a 3 minutes forecasting horizon.

As explained before, only one day was chosen to demonstrate the methodology developed. Although it may seem a very small sample, since we picked a day with a very high irradiance variability, this day is actually more relevant than if we would choose many days including totally clouded or clear days, which would

essentially have zero errors.

The errors when identifying clouds can come from many sources:

- Thin clouds - Thin clouds, like *cirrus* ou *cirrostratus* cause a decrease in DNI around 20%. Therefore, they cause errors if identified as clouds or if not identified as clouds, influencing all time horizons. A solution to this problem may be use an intermediate cloud identification, distinguishing between thin and thick clouds. However, this may have the problem of different cloud velocities.
- Formation and evaporation of clouds - This problem is even more difficult to overcome and is specially notorious in coastal areas, like ours. These errors influence the most long temporal horizons. To date, there isn't any method in the literature capable of accounting for this phenomena.
- Circumsolar region - cloud decision in the circumsolar region is, by far, the one that incurs in larger errors. The problems contributing to this are the solar obstructor and the smearing and blooming effect. Without the solar obstructor, the latter effects would be even worse. Hence, it's very difficult to beat the persistence model for very short time horizons. However, these problems do not affect larger time horizons.
- Perspective errors - although not an obvious error source, perspective errors can lead to very large errors for longer time horizons. Due to the fact that images are in 2D, pixels identified as clouds in the outer regions of the image are usually over estimated, since the side of the clouds is misinterpreted as its bottom. Furthermore, a cloud can hide another and both are perceived as one, even if there is a lot of clear sky between them. A possible solution to this problem would be using different SI's covering a larger spatial horizon.

Errors associated with the velocity vectors are the following:

- Heterogeneous velocity field - When clouds move in different directions usually when we have a difference in the altitude of the clouds), the representative vector is the average of both, resulting in very large errors, specially to longer time horizons. A possible solution for this is to identify only one of the kind of clouds and perform the CCM only on that region. Another possible solution, which would imply more computational power, would be to use many different vectors for the same image, instead of only one representative vector.
- Rotation of the clouds - even with an homogeneous velocity field, the clouds may present rotational behavior, instead of linear. Since our model assumes linearity, it would incur in very large errors. However, this is not very common.

Another possible source of error, exclusive of our SI, may happen if the clouds consistently hide behind the obstructor, with a velocity field in the direction of the pole. Finally, some of the errors may have an origin on the measured irradiance values themselves, since the PV panels used are not an ideal way to measure DNI.

5. Conclusion and future work

A SI prototype was built from the ground up, based on an innovative geometry. The cost of the prototype was kept under 300€, way below any value found in the literature. Based on this SI, we developed a method for predicting DNI for a short time horizon, an important component of solar energy industry.

Even though this project is still in its initial stage, we found consistent improvements in our methods, when comparing with the persistence model for time horizons over 1 minute. For shorter periods, the solar obstructor and the circumsolar glare don't allow for a good prediction. Still, we achieved good results for 2 and 3 minute horizons, due to the small size of the obstructor, contrary to other works found in the literature.

The major problems regarding the prototype are its low motor precision and the limited FOV of the camera. Both are easily solved, specially by implementing a reduction gearbox for the motor and an hemispherical lens for the camera. Increasing the FOV would augment the spatial resolution and, consequently, the time horizon available. Improving the precision of the motor would allow the camera to follow the Sun more frequently, reducing even more the obstructor dimensions, and thus allowing for shorter time horizons.

Regarding the software, we suggest the recognition of 2 distinct types of clouds (thin and opaque) when calculating the cloud coverage, as well as the application of CCM only to the region identified as clouds. This a significant topic, as thin clouds were one of the major sources of error in this work.

To validate more consistently the results obtained, a method for measuring DNI without associated errors and with a bigger acquisition frequency would be necessary, as the PV panels used here were not designed for that.

References

- [1] P. Denholm and R. M. Margolis. Evaluating the limits of solar photovoltaics (pv) in electric power systems utilizing energy storage and other enabling technologies. *Energy Policy*, 35(9):4424–4433, Abril 2007.
- [2] L. E. Jones. Strategies and decision support systems for integrating variable energy resources in control centers for reliable grid operations - global best practices, examples of excellence and lessons learned, Dezembro 2011. Report under contract no DE-EE0001375 of U.S. Government.
- [3] E. Lorenz, J. Hurka, D. Heinemann, and H. Beyer. Irradiance forecasting for the power prediction of grid-connected photovoltaic systems. *Selected*

- [4] R. Perez, S. Kivalov, J. Schlemmer, Hemker Jr. K., D. Renne, and T. Hoff. Validation of short and medium term operational solar radiation forecasts in the us. *Solar Energy*, 84:2161–2172, 2010.
- [5] P. Mathiesen and J. Kleissl. Evaluation of numerical weather prediction for intra-day solar forecasting in the conus. *Solar Energy*, 85(5):967–977, 2011.
- [6] R. Marquez and C.F.M. Coimbra. Forecasting of global and direct solar irradiance using stochastic learning methods, ground experiments and the nws database. *Solar Energy*, 85(5):746756, 2011.
- [7] M. Noia, C. Ratto, and R. Festa. Solar irradiance estimation from geostationary satellite data: Ii. physical models. *Solar Energy*, 51(6):457465, 1993.
- [8] A. Hammer, D. Heinemann, E. Lorenz, and B.L. Ckehe. Short-term forecasting of solar radiation: a statistical approach using satellite data. *Solar Energy*, 67:139150, 1999.
- [9] C. W. Chow, B. Urquhart, M. Lave, A. Dominguez, J. Kleissl, J. Shields, and B. Washom. Intra-hour forecasting with a total sky imager at the uc san diego solar energy testbed. *Solar Energy*, 85(11):2881–2893, 2011.
- [10] J. E. Shields, R. W. Johnson, and T. L. Koehler. Automated whole sky imaging systems for cloud field assessment. In *Fourth Symposium on Global Change Studies*. American Meteorological Society, 1993.
- [11] M. Souza-Echer, E.B. Pereira, L. Bins, and M. Andrade. A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera. *J. Atmos. Ocean. Technol.*, 23:437–447, 2006.
- [12] A. Cazorla, F. J. Olmo, and L. Alados-Arboledas. Development of a sky imager for cloud cover assessment. *J. Opt. Soc. Am*, 25:29–39, 2006.
- [13] Q. Li, W. Lu, and J. Yang. A hybrid thresholding algorithm for cloud detection on ground-based color images. *Journal of Atmospheric and Oceanic Technology*, 28 Oct:1286–1296, 2011.
- [14] M.S. Ghonima, B. Urquhart, C.W. Chow, J.E. Shields, A. Cazorla, and J. Kleissl. A method for cloud detection and opacity classification based on ground based sky imagery. *Atmos. Meas. Tech.*, 5:2881–2892, 2012.
- [15] J. Shields, M. Karr, A. Burden, R. Johnson, V. Mikuls, J. Streeter, and W. Hodgkiss. Research toward multi-site characterization of sky obscuration by clouds. *Grant N00244-07-1-009*, Technical Note 274, 2009.
- [16] T. Hamill and T. Nehr Korn. A short-term cloud forecast scheme using cross correlation. *Weather Forecasting*, 8:401–411, 1993.
- [17] C.J. Long, J. Sabburg, J. Calbo, and D. Page. Retrieving cloud characteristics from ground based daytime color all-sky images. *J. Atmos. Ocean. Technol.*, 23:633–652, 2006.
- [18] H. Yang, B. Kurtz, D. Nguyen, B. Urquhart, C. W. Chow, M. Ghonima, and J. Kleissl. Solar irradiance forecasting using a ground-based sky imager developed at uc san diego. *Solar Energy*, 103:502–524, 2014.
- [19] GoPro. *Hero3 White Edition User Manual*.
- [20] Emg30, mounting bracket and wheel specification. <http://www.robot-electronics.co.uk/html/emg30.htm>. [Online; acedido a 25-Mar-2015].
- [21] R. Paleta, A. Pina, and C.A.S. Silva. Polygeneration energy container: Designing and testing energy services for remote developing communities. *IEEE TRANSACTIONS ON SUSTAINABLE ENERGY*, 5(4), 2014.
- [22] Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html. [Online; acedido a 25-Mar-2015].
- [23] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1(c), 1999.
- [24] J. Heikkil and O. Silven. A four-step camera calibration procedure with implicit image correction. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.
- [25] Rodrigues’ rotation formula. <http://mathworld.wolfram.com/RodriguesRotationFormula.html>. [Online; acedido a 25-Mar-2015].
- [26] I. Reda and A. Andreas. Solar position algorithm for solar radiation application. *National Renewable Energy Laboratory*, 560(34302), 2003.
- [27] N. Mori and K.A. Chang. *Introduction to MPIV - PIV toolbox in MATLAB, version 0.97*, Julho 2009.
- [28] S. Bansal and D. Aggarwal. Color image segmentation using cielab color space. *IJCSET*, 1(415-420), 2011.