

Article

Solar Irradiance Forecast Based on Cloud Movement Prediction

Aleksander Radovan ^{1,*} , Viktor Šunde ² , Danijel Kučak ³ and Željko Ban ⁴¹ BISS Ltd., 10000 Zagreb, Croatia² Department of Electric Machines, Drives and Automation, Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia; viktor.sunde@fer.hr³ Department of Software Engineering, Algebra University College, 10000 Zagreb, Croatia; danijel.kucak@algebra.hr⁴ Department of Control and Computer Engineering, Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia; zeljko.ban@fer.hr

* Correspondence: aleksander.radovan@fer.hr; Tel.: +385-98-637-784

Abstract: Solar energy production based on a photovoltaic system is closely related to solar irradiance. Therefore, the planning of production is based on the prediction of solar irradiance. The optimal use of different energy storage systems requires an accurate prediction of solar irradiation with at least an hourly time horizon. In this work, a solar irradiance prediction method is developed based on the prediction of solar shading by clouds. The method is based on determining the current cloud position and estimating the velocity from a sequence of multiple images taken with a 180-degree wide-angle camera with a resolution of 5 s. The cloud positions for the next hour interval are calculated from the estimated current cloud position and velocity. Based on the cloud position, the percentage of solar overshadowing by clouds is determined, i.e., the solar overshadowing curve for the next hour interval is calculated. The solar irradiance is determined by normalizing the percentage of the solar unshadowing curve to the mean value of the irradiance predicted by the hydrometeorological institute for that hourly interval. Image processing for cloud detection and localization is performed using a computer vision library and the Java programming language. The algorithm developed in this work leads to improved accuracy and resolution of irradiance prediction for the next hour interval. The predicted irradiance curve can be used as a predicted reference for solar energy production in energy storage system optimization.

Keywords: solar irradiance prediction; cloud movement; sky image processing



Citation: Radovan, A.; Šunde, V.; Kučak, D.; Ban, Ž. Solar Irradiance Forecast Based on Cloud Movement Prediction. *Energies* **2021**, *14*, 3775. <https://doi.org/10.3390/en14133775>

Academic Editor: Dinko Vukadinović

Received: 17 May 2021
Accepted: 22 June 2021
Published: 23 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In this paper, a prediction of solar irradiance was proposed based on a sky image that includes the sky, clouds and their movement, and sun coverage. The motivation for this research was to obtain a solar-based virtual power plant. In such a plant, the maximum production limit curve for a future period could be obtained from the long-term hourly irradiance forecast of the hydrometeorological institute. The production limit curve is essential for the power system dispatcher to ensure the stability of the power system. The irradiance forecast represents the average hourly irradiance. In order to achieve the planned average power generation, it is necessary to use energy storage to compensate for the deviation of the current irradiance from the average. Optimal use of energy storage devices with different efficiency, capacity, and response times can be achieved if the more accurate irradiance prediction for the time interval of the next hour is available.

The 1-h horizon forecast elaborated in this paper is based on cloud motion prediction and a wide-angle ground camera that takes a picture of the sky every 5 s. The direction of cloud movement is calculated from the movement of their centroids and the detected cloud edges within 1 min (after taking 12 consecutive images of the sky). Based on the visible clouds in the sky and their movement, the prediction of the clouds, the position of the sun, and the prediction of the solar irradiance are performed. The described method

has proven to be simple since the shape of the clouds and the sun is approximated by a rectangular shape based on only four points, which does not depend on sophisticated algorithms and does not require the training of neural network models based on large data sets. The strength and novelty of this research are in its simplicity (it approximates the shape of the clouds with rectangles) that makes it practical and performant (the movement simulation of clouds and sun is based on moving only four points that define the rectangle shape). Additionally, it can be used in a virtual plant environment based on only one commercial camera and with the prediction of sun coverage level with clouds during the following hour period. Therefore, there is no need to use satellite images of the sky and gather data for training models based on machine learning.

Prediction of solar radiation has been the subject of many studies in recent years. Some studies have introduced artificial neural networks or deep learning with extensive data sets. Examples of solar radiation prediction based on machine learning [1–6] showed that solar radiation prediction could be affected by different input parameters for artificial neural networks and multilayer perceptrons. The other machine learning method based on publicly available weather reports showed the approach for a prediction horizon of 24 h [7]. One of the methods described a prediction accuracy of 2 days in advance based on a dataset covering a period of 8 months [8]. These methods allow the development of a system that operates autonomously but depends on collected data and the training of a model that must be retrained as parameters change over time. Moreover, if the prediction period is too long, the approach is not suitable for microgrid systems that support different types of energy storage [9–11] based on dynamic adaptation and parameters of the system based on capacity, current load, and speed of energy storage. Compared to the method described in this paper, machine learning-based systems require a large set of historical data and time to train the models to make the prediction. Furthermore, the need to retrain the model after collecting new datasets after deploying the production system to improve it does not exist. The model described in this paper does not depend on existing historical data.

Other methods used cloud motion prediction based on satellite images and motion vectors to predict output power, but only 30 min in advance, which may be a limitation, especially when some power storage devices in the microgrid system need more time to change their operating mode from charging to discharging [12]. A similar method has been used with several fisheye cameras but could predict up to 15 to 30 min of updates per minute [13], but our method manages to predict solar coverage up to 1 h.

The development of cloud detection algorithms based on sky images has also received many variants. One of the approaches analyzed the way of reducing sunlight interference in ground-based sky images [14]. It has been shown that HSV color space gives the best results, so the same approach is used in this work to accurately detect the cloud edges, cloud centroid, and cloud motion. Optimal HSV thresholds for best results in cloud edge detection can be determined by an artificial neural network-based system described in our previous work [15] since the sky image often contains different parameters (number of clouds, cloud sizes, and histogram values) and colors that may require different settings for image processing. Other variants of cloud detection on sky/cloud images captured by ground-based cameras are based on cloud segmentation [16], which is learning-based and not suitable for rapidly changing weather conditions. Some research involved artificial neural networks that depend on the datasets for a given area [17,18]. These approaches are only suitable for specific areas and often cannot be applied to all locations.

An Internet of Things-based solution for intra-minute cloud passing forecasting that uses the cloud's center of gravity can also be used to control the reduction of the output power of photovoltaic systems [19]. However, the proposed solution is focused on the short-term prediction of the moment when the clouds start to cover the sun, which may be too short to effectively control different types of energy storage devices in a microgrid system.

Satellite images, together with a support vector machine learning scheme, can also be used for solar power prediction as well [20], but the approach is not suitable for solar

systems that are located at a specific geographical location and depends on the position of the clouds and the sun.

Cloud motion estimation based on a block matching algorithm performs the motion of a pixel block over a region and searches for two blocks in two consecutive images that have the highest degree of similarity [21], which proves that using sky images is a reliable method for cloud motion prediction. The only challenge with this approach is the longer processing time, making it challenging to use under real-time conditions.

This work is based on the assumption that the long-term irradiance forecast with hourly resolution from the hydrometeorological institute is accurate enough to be used as an upper bound reference for scheduling energy production for the power system dispatcher. To obtain the declared constant hourly interval production equal to the declared possible production based on the average hourly estimates, it is necessary to use an energy storage system. If the energy storage system consists of different storages with different capacities, efficiency, and response speed, the efficient use of these storages is possible if an accurate forecast of irradiation is available with a maximum of 1 min time sample in a 1-h forecast window. For this purpose, it is necessary to obtain a curve of the percentage of the uncovered sun with clouds for 1 h in the future. Normalizing the mean of this curve to the value predicted by the hydrometeorological institute should give the accurate curve of irradiance for that time. The curve of predicted irradiance, corrected with coefficients characteristic of the production facility, could serve as a basis for online optimization of the use of the power system. The solar coverage prediction is based on wide-angle camera images of the sky.

The possible utilization of this prediction model and the generated curve of predicted solar irradiance is in a virtual plant that consists of different storages and is based on integration with solar photovoltaic cells. Storages can be based on a battery [22], hydrogen cell [23], and supercapacitor [24], each with different characteristics related to capacity, efficiency, and response speed. With the generated curve of predicted solar irradiance in the next hour, the amount of energy generated can be predicted. Depending on the energy storage condition in terms of their charge, efficiency, and response speed, it can be determined which storage should be used to store generated energy. If the sun is visible in the next period, the solar energy will be generated and stored in one of the storages, or if the sun is not visible, then the stored energy needs to be consumed. Among different storages, optimizing energy flows ensures the use maximization of stored energy [25].

Section 2 describes the master control algorithm of sun un-coverage forecast in the hybrid storage system. It contains a flow diagram of the algorithm with a description of the sequence of steps.

The algorithm for detecting the sun and clouds is obtained in Section 3, where camera calibration is described and contour and centroid detection of both the sun and clouds.

The list of clouds that are candidates for sun coverage according to their current position and velocity vector is obtained in Section 4. The irradiance curve is obtained in the same section using subsequent calculation of cloud position and sun coverage with a determined time sample.

The results of sun un-coverage percentage prediction and real-time un-coverage percentage comparison are presented in Section 5, while the concluding considerations and list of references are given in Section 6, respectively.

2. Master Control Algorithm of Sun Un-Coverage Forecast in a Hybrid Storage System

2.1. Hybrid Storage System

Solar-based virtual power plant combines different types of storage systems with different efficiencies, capacities, and response times. Figure 1 shows an example of such types of storage systems connected to a photovoltaic system.

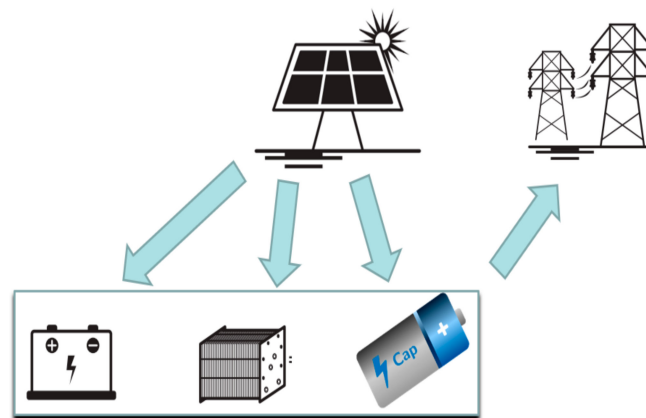


Figure 1. Hybrid storage system.

It is important for the power system dispatcher to ensure the stability of the power system. In order to achieve the average value of energy production, it is necessary to use energy storage to compensate for the deviation of the current irradiance from the average value. Optimal use of energy storage can be achieved if a more accurate prediction of irradiance during the time interval of the next hour is available.

Optimal use of various energy storage systems requires accurate prediction of solar irradiance with at least an hourly time horizon. To achieve this, a cloud movement prediction algorithm must be used to generate the curve indicating the sun un-coverage level in the next period and, based on this, predict the solar irradiance in the next hour. Using the long-term hourly irradiance forecast of the hydrometeorological institute, the direct solar irradiance can be determined. In the next sections, the algorithm consisting of the calibration of the ground-based sky camera, detection of the contours of clouds and the sun on the captured images, the prediction of their movement, and the generation of the sun un-coverage curve generation is described.

2.2. Master Control Algorithm

Figure 2 shows the main algorithm, which includes image acquisition, image processing, simulation of solar coverage, and generation of the curve showing the degree of solar coverage in the next period. It contains two sub-algorithms, one of which focuses on image processing (described in Section 3) and the other on solar coverage estimation (described in Section 4).

The methodology used for contour and centroid detection is described in Section 3. To simplify and speed up the calculations, each contour of the clouds and the sun is approximated by a rectangular shape that can be easily shifted to the predicted positions by adding the shift value to the x and y coordinates of the points defining the rectangular shape. To further optimize the algorithm, only the cloud candidates that can theoretically cover the sun are considered. The selection of clouds is based on their position (all clouds whose position and direction cannot lead to the coverage of the sun are ignored). The cloud candidates that can cover the sun are stored for further processing.

As shown in Figure 2, before the first sub-algorithm, “Applying image processing algorithm” starts, the images need to be prepared and ready for processing. The results generated by the image processing algorithm, processed images with detected sun and clouds, are then used to start the second sub-algorithm, “Applying Sun coverage estimation algorithm” that generated a CSV file with data needed for developing the curve of uncovered sun.

In the next section, the clouds and sun detection algorithm will be described, from the first stage that removes distortion effect on images generated by wide-angle camera GoPro Fusion and detection of cloud and sun contours with JavaCV library.

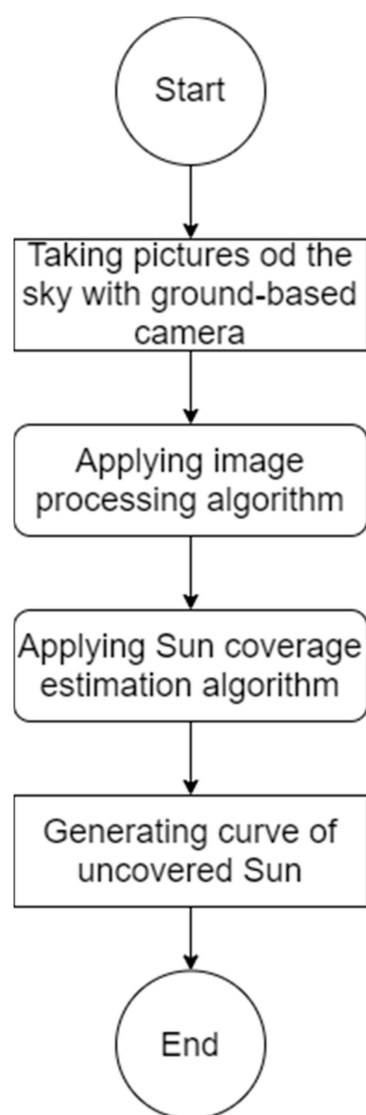


Figure 2. Master algorithm flow diagram.

3. Clouds and Sun Detection Algorithm

Ground-based sky images were captured from the site in Zagreb, Croatia, using the GoPro Fusion wide-angle camera with authors as photographers during the experiment. The sky images for the experiment were captured with a sampling time of 5 s, starting on 12 July 2020 at 10:28:50 h until 15 July 2020 08:25:20 h. Figure 1 shows few examples of captured images that are cloud-free or contain different types of clouds. The east is on the left side of the images, the west on the right side, the north at the bottom of the image, and the south at the top of the image.

The image in Figure 3a shows an example of an image with a clear sky without clouds (it was taken on 13 July 2020 at 08:09:09), Figure 3b an image with many cumulus clouds (it was taken on 13. July 2020 at 13:26:20), Figure 3c an image with cirrus clouds (it was taken on July 14 at 09:09:15), and Figure 3d an image taken at the end of the day when the sun was setting over the horizon (it was taken on July 14 at 18:45:32).

From the images in Figure 3, it can be seen that the wide-angle camera used distorts the image quite a bit. In order to obtain a linear motion of the cloud according to its velocity vector, it is necessary to undistort the image.

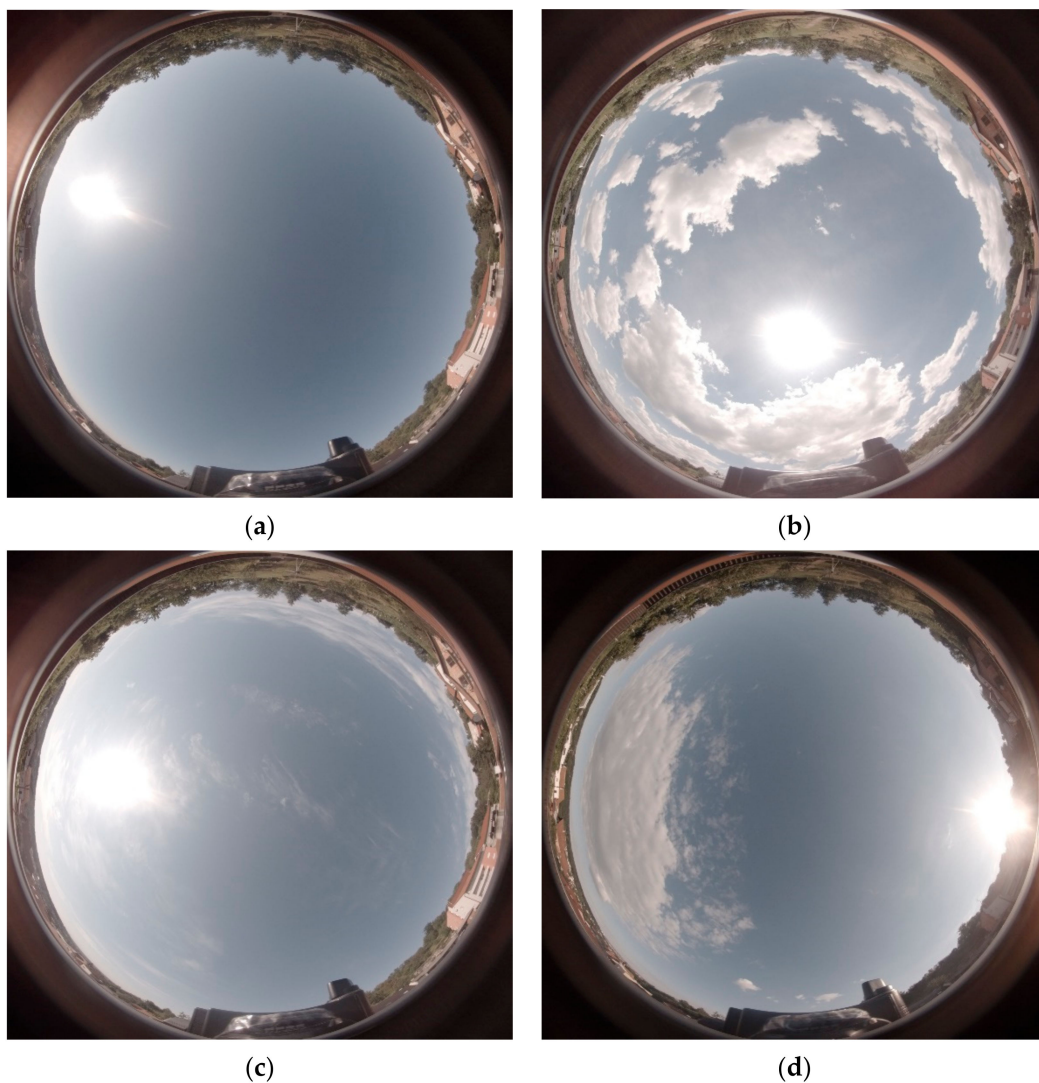


Figure 3. Ground-based sky images.

3.1. Camera Calibration

In order to correct the distorted image of the GoPro Fusion camera and define the parameters of the undistortion algorithm, a camera calibration was performed. For this purpose, the image of the checkerboard $100\text{ cm} \times 90\text{ cm}$ with the size of the squares $2\text{ cm} \times 2\text{ cm}$ is used.

Omni directional Camera Calibration Toolbox for Matlab [26–28] is used for image undistortion. It allows the user to calibrate any omnidirectional camera in two steps: collecting multiple images of a checkerboard shown in different positions and orientations and extracting the vertices. After calibration, it provides functionalities to express the relationship between a given pixel point and its projection on the unit sphere. The calibration procedure starts with the selection of images with checkerboard patterns and continues with the detection of the checkerboard points using the method “detectCheckerboardPoints”. The image size is determined based on the first image read. In our case, the image resolution was 3104×3000 pixels. In the next step, the square size of the checkerboard is determined and the world coordinates of the corners of the squares are generated using the method “generateCheckerboardPoints”. After that, the calibration is performed based on the fisheye parameters using the method “estimateFisheyeParameters”. The last phase is the undistortion of the image using the method “undistortFisheyeImage”. Figure 4a shows the distorted checkerboard used for camera calibration, and Figure 4b shows the

undistorted version after the described procedure has been performed. Figure 4d,e show the distorted and undistorted version of the image with the clouds and the sun located in the center of the image, Figure 4f,g show the distorted and undistorted version of the image with the sun located on the left side of the image (near the east). Figure 4c shows the Matlab generated visualization of the extrinsic parameters after running the “show Extrinsic” function. It shows the rendered 3-D visualization of the extrinsic parameters by viewing the calibration patterns with respect to the camera. It plots the locations of the calibration pattern in the camera’s coordinate system that were taken during the calibration phase. Each enumerated location represents a different angle, position, and distance of the checkerboard from the camera taken during the calibration time. The location of the camera is shown in dark blue, and the positions of the board with checkerboards are shown in other colors in Figure 4c. Different colors are used to make the image distinguishable more easily. Only those images with checkerboards are considered valid that contain a complete visible board [29].

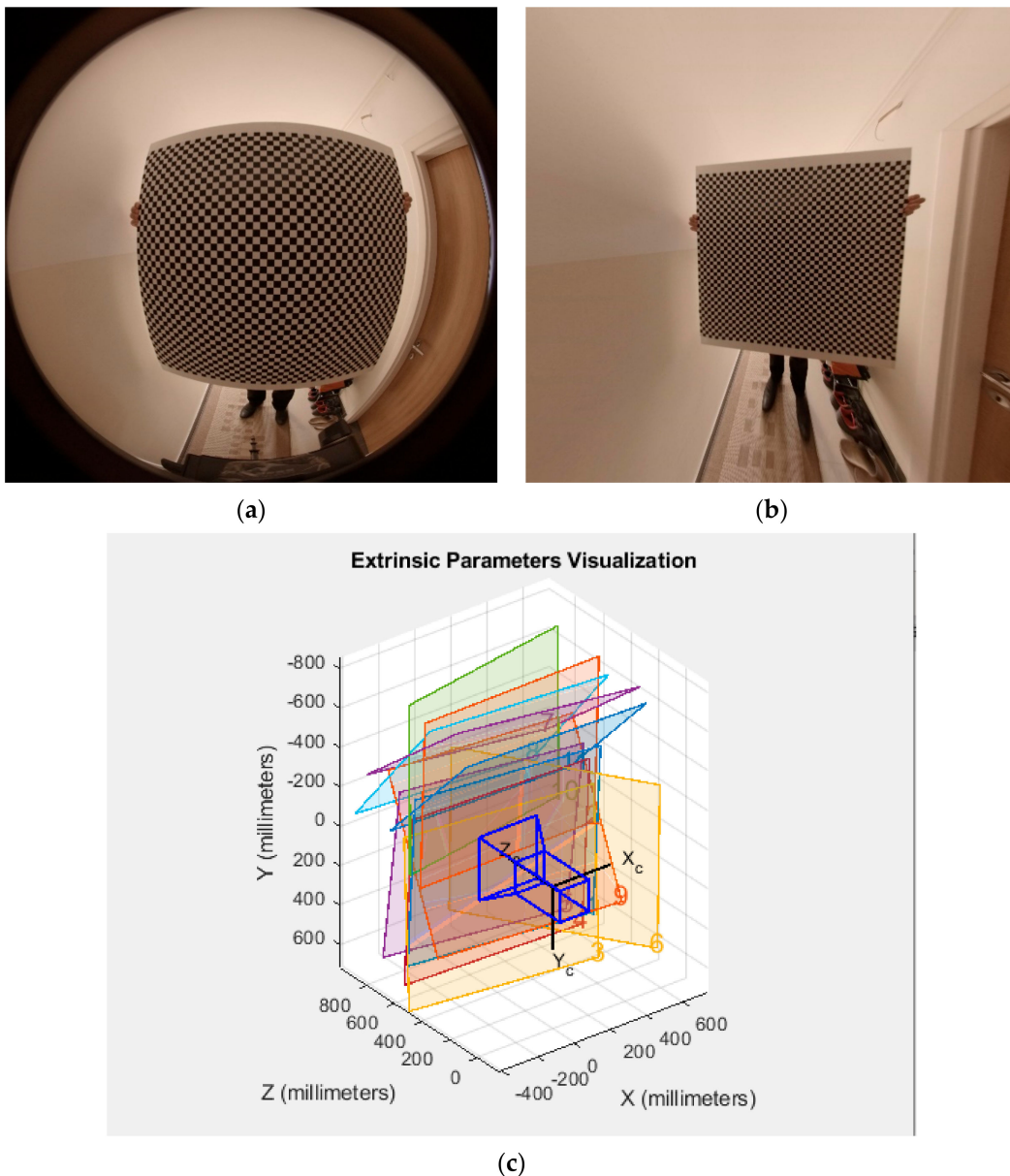


Figure 4. Cont.

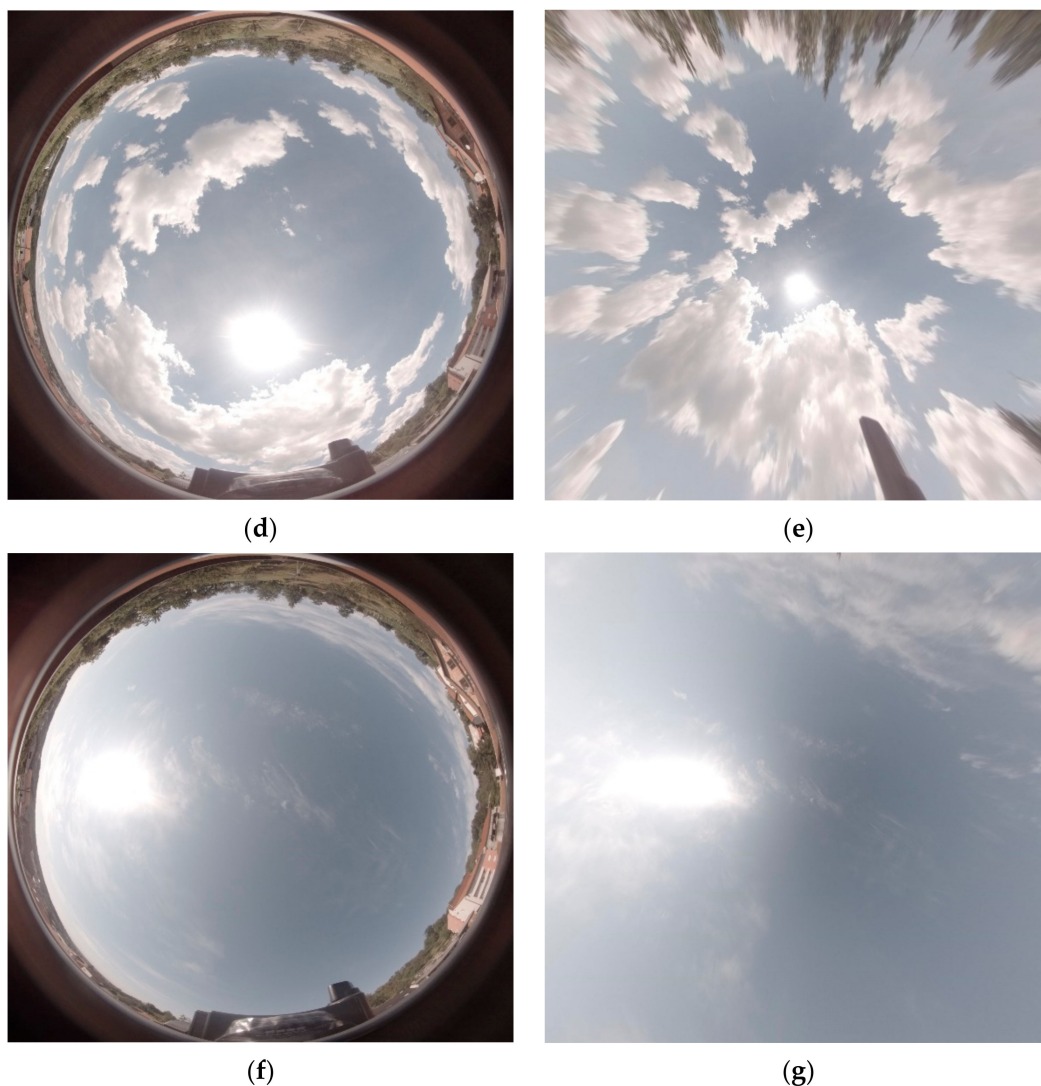


Figure 4. Image transformation of GoPro Fusion camera calibration: (a) distorted checkerboard image, (b) undistorted checkerboard image, (c) Matlab-generated visualization of extrinsic parameters, (d) distorted image with the sun at the center of the image, (e) undistorted image with the sun at the center of the image, (f) distorted image with the sun at the left (east) side of the image, and (g) undistorted image with the sun at the left (east) side of the image.

The obtained undistorted images allow the calculation of the cloud position and velocity.

After undistorted images of the sky were obtained, the next step is to recognize the contour and center of gravity of the sun, described in the following section.

3.2. Recognition of the Contours and the Center of Gravity of the Sun

The flowchart of the image processing algorithm mentioned in the previous section is shown in Figure 5.

The first phase of the algorithm used is to undistort the image and then detect the contour of the sun, the centroid of the sun contour, and then the contours of all the clouds in the image and their corresponding centroids. To improve the detection results, only significant areas of the sun and cloud contours are considered. For the sun contour, the range of 5000 pixels is used as the threshold for detecting the sun. When a cloud starts to partially cover the sun, the area of the sun shrinks, and when the area is below the aforementioned 5000 pixels, the sun is considered to be covered by clouds. The threshold is based on the camera resolution (3104×3000 pixels) and the visible area of the sun in the clear or uncovered sky, as shown in Figure 6c. It is determined by the calculated area of the

sun during the period of taking pictures of the sky and using a minimum value of the area when the sun is visible and not covered by clouds.

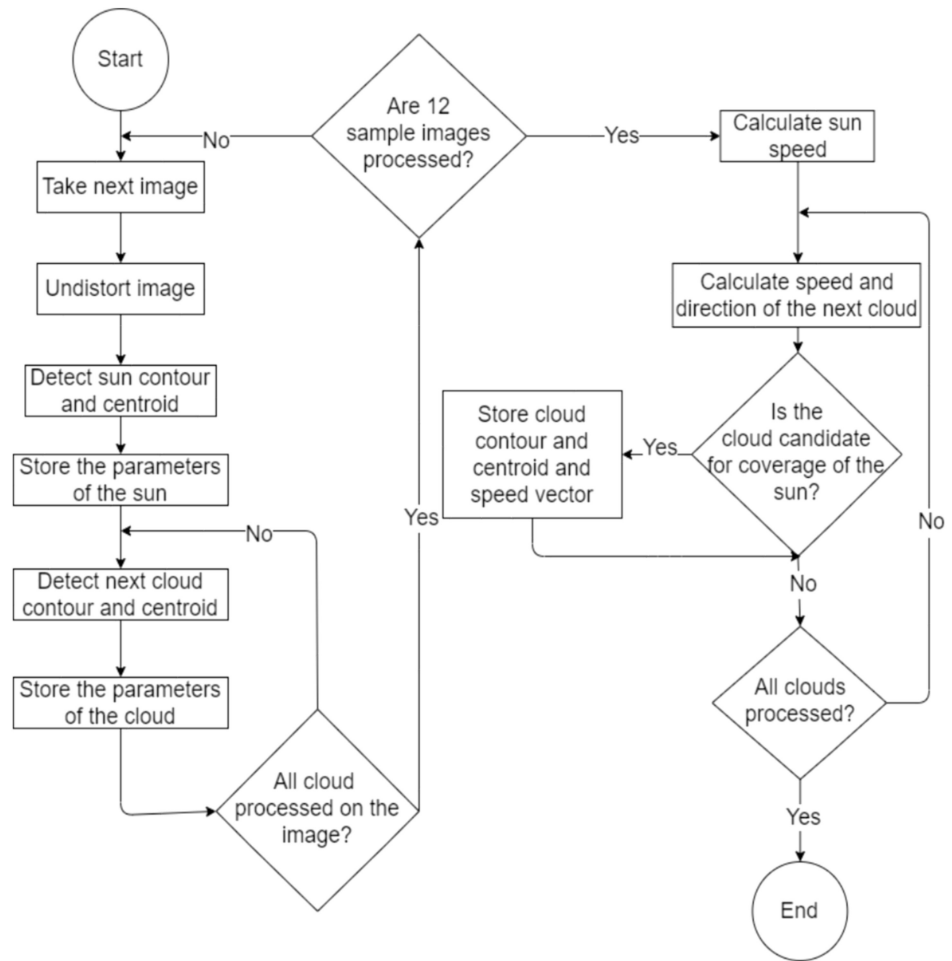


Figure 5. Image processing algorithm flow diagram.



(a)



(b)

Figure 6. Cont.

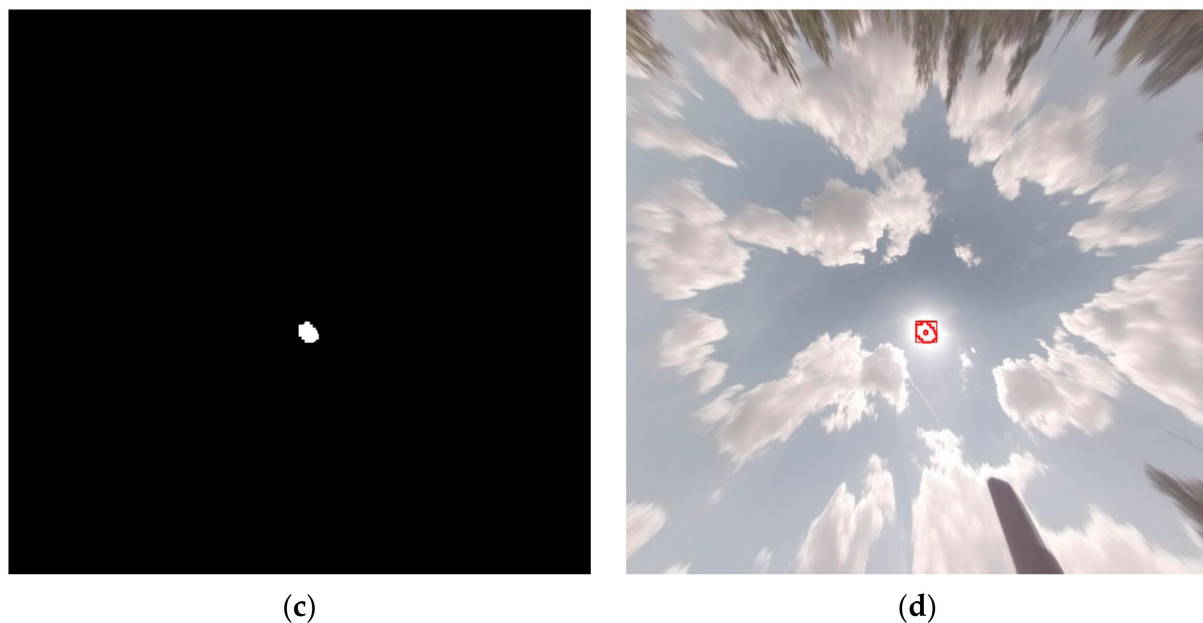


Figure 6. Image processing process for sun detection: (a) original image, (b) HSV format of the image, (c) detected sun after erosion operation, and (d) detected sun contour with rectangular shape approximation.

As with the sun area threshold, only clouds with the significant area are considered since those with too small of an area do not significantly block the sun. The threshold for cloud area is set to 3000 pixels: clouds with an area less than 3000 pixels are simply ignored and not stored in the ArrayList structure in the Java programming language with other clouds that may significantly cover the sun. The cloud area threshold is based on the sun area threshold: if the calculated cloud area is 60% of the calculated sun area, it is considered a significant coverage level.

The first step is to detect the sun in the sky by performing morphological operations of erosion. To detect the sun, the following image processing operations must be performed first (as described in our previous work [30]): converting the image from RGB (Red, Green, and Blue) to HSV (Hue, Saturation, and Value) color model, setting the size for structured elements for erosion, finding contours on the image that represent the sun, and finding the sun's centroid.

Morphological operations on a binary image create a new binary image in which the pixels have a non-zero value only if the test succeeds at the location in the image. A structuring element is a binary image represented by a matrix of pixels that have values of zero or one. The dimensions of the matrix indicate the size of the structuring element. The erosion of a binary image is described with the following equation:

$$n = i \ominus s \quad (1)$$

where n represents the new binary image after erosion operation, i is the initial binary image that was transformed, and s represents the structuring element.

In the resulting image, there are ones at the coordinates (x, y) of the origin of a structuring element s where it matches the input image i , $n(x, y) = 1$ if the structuring element s matches i and zero otherwise. In the case of sun detection, erosion removes clouds from a binary image, and in the case of cloud detection, it removes very small clouds and reduces the size of clouds to emphasize the boundaries between clouds. The dilation operation has the opposite effect on erosion: it adds a layer of pixels to both the inner and outer boundaries of the contours.

The conversion from the RGB color model to the HSV color model can be done with the JavaCV library (based on the OpenCV library written in C++) using the method "cvtColor" from the class "Imgproc". Since the images containing the sun consist of

glittering segments, the minimum and maximum values for the H and S components can be set to 0 (this represents the filter that suppresses the H and S components), but the V component is set from a minimum value of 240 to a maximum value of 255. Figure 6a shows the initial RGB image and the resulting HSV image afterward in Figure 6b.

The structuring element used in erosion was set at the size of 50×50 pixels. The size was determined based on the image resolution (3104×3000 pixels) and the criteria for detecting the most glittering element on the image, namely the sun. Figure 5 shows the steps that were performed to detect the sun: Figure 6a shows the original image, Figure 6b shows the sky image converted to HSV format using the minimum HSV components (0, 0, 240) and the maximum values (0, 0, 255), Figure 6c shows the image after the erosion process (the white part represents the detected sun), and Figure 6d shows the image with the detected contour, the centroid point as well as the rectangular approximation to the shape of the sun contour (drawn in red). The H and S components of the HSV minimum and maximum values are set to zero to ignore the color component portion of the model (with H component) and the amount of gray (with S component). The value for the V component is the only component used because it describes the brightness of the color. The minimum and maximum values, set to 240 and 255, respectively, give the best results for detecting the most glittering element in the sky, namely the sun.

After detecting the sun and applying the erosion operation, the only contour detected on the image was the contour of the sun. Using the “findContours” method available in the “Imgproc” class of JavaCV, the sun contour is accurately detected and drawn using the “drawContours” method in Figure 6d. In addition to the sun contour, a minimum-area bounding rectangle enclosing the contour points is found using the “minAreaRect” method from the “Imgproc” class in the same figure.

As clouds are moving over the horizon, the sun is moving as well, although much slower than the clouds. To detect the sun movement, the centroid of the sun needs to be detected. Since the images were taken every 5 s, the sun movements are often very small, but the sun travels over the whole sky during the day. The sun’s centroids are calculated as the arithmetic mean of all the contour points as shown in Equation (2):

$$c = \frac{1}{n} \sum_{i=0}^n (x_i, y_i) \quad (2)$$

where c is the centroid of the sun, n is the number of points in the contour, and (x_i, y_i) are the coordinates of the points in the contour.

To find the center of gravity of the sun in JavaCV, moments must be used. A moment is a certain weighted average of the intensities of the image pixels that can be used to find some specific properties of an image, such as the centroid. In the case of a raster image, the spatial moments are calculated as [31] shown in Equation (3). The central moments are calculated using Equation (4), where (\hat{x}, \hat{y}) is the centroid calculated using Equations (5) and (6).

$$m_{ji} = \sum_{x,y} (array(x, y) \cdot x^j \cdot y^i) \quad (3)$$

$$mu_{ji} = \sum_{x,y} (array(x, y) \cdot (x - \bar{x})^j \cdot (y - \bar{y})^i) \quad (4)$$

$$\hat{x} = \frac{m_{10}}{m_{00}} \quad (5)$$

$$\hat{y} = \frac{m_{01}}{m_{00}} \quad (6)$$

where m_{ji} represents the spatial moment, $array(x, y)$ is the array of 2D points of the image, x^j and y^i are coordinates of points, mu_{ji} is the central moments, and (\hat{x}, \hat{y}) represents the center of mass. Figure 6d shows the detected centroid of the sun (the red circle inside of the sun contour).

3.3. Detection of the Contours and the Centroid of the Sun

After successfully locating the sun and detecting the centroid with contour, the next step was to detect clouds on the image. The process of cloud edge detection consists of converting the color model, applying the morphological operations erosion and dilation, and shape approximation with the bounding rectangle with minimum area. The images generated during the process are shown in Figure 7.

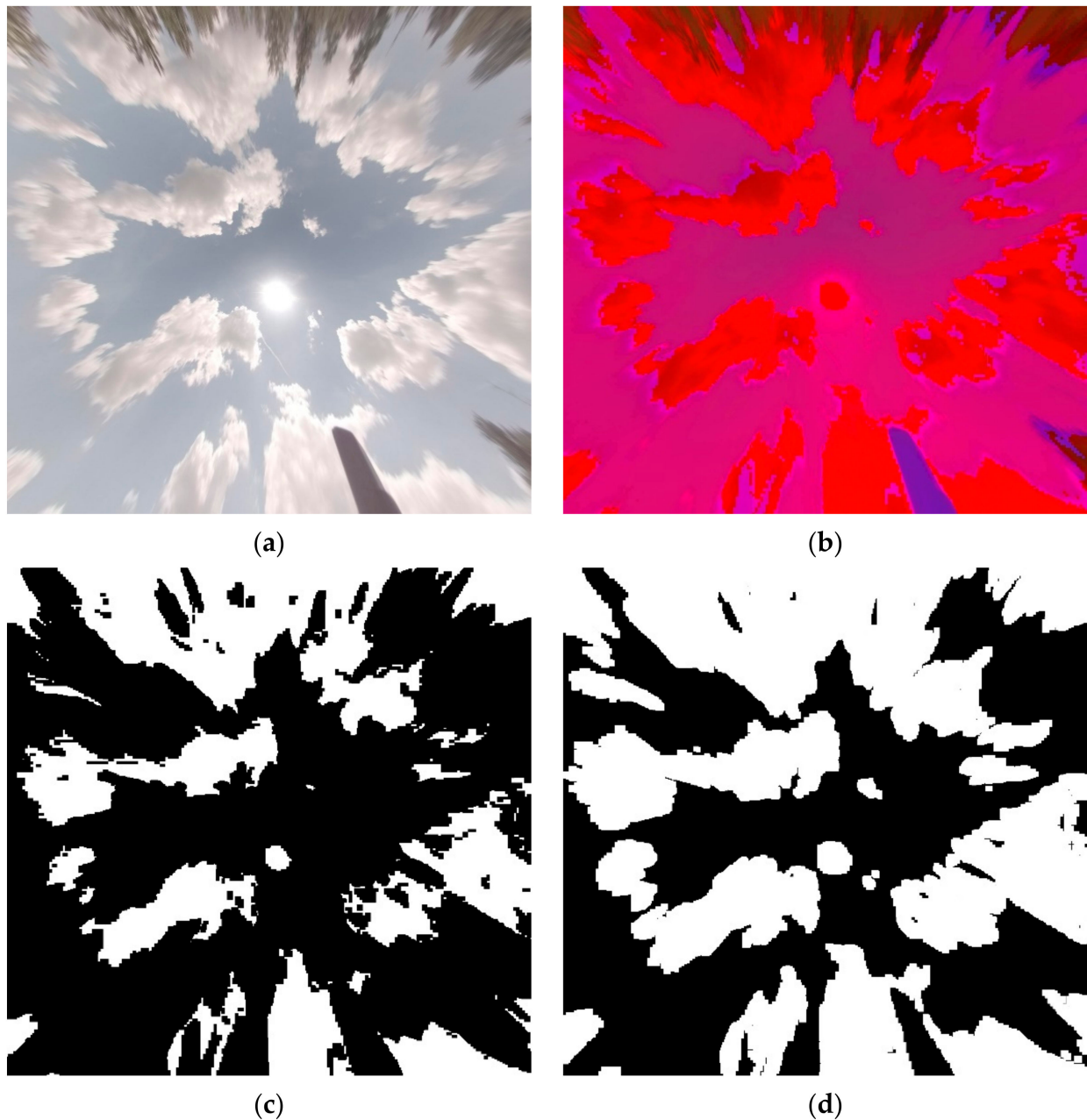


Figure 7. Image processing process for cloud detection: (a) original image, (b) HSV format of the image, (c) detected clouds after erosion operation, and (d) detected clouds after dilation operation.

The minimum and maximum values for the HSV components were set to (0, 0, 0) and (70, 70, 255) to obtain the best results. Compared to the parameters used for sun detection, where only the most glittering element on the image was in focus, cloud detection also focused on other colors, such as different shades of gray. The H-component is the color component of the model, and with values between 0 and 70, the dominant color is red (Figure 7b). The S component describes the amount of gray in a given color, and reducing

this component results in a more gray color, which is a very common color of clouds. The V component is not filtered, covers the entire range from 0 to 255, and in conjunction with saturation, describes the brightness of the color to cover all shades of gray. The original image is shown in Figure 7a, and the image transformed to the HSV color model is shown in Figure 7b.

The size of the structuring element for the morphological operation of erosion was set to the size of 20×20 pixels, and it was determined using the image resolution (3104×3000 pixels) and the criteria for the removal of very small clouds (with the calculated area below 3000 pixels). The resulting image after transformation is shown in Figure 7c. In order to obtain the best results in cloud edge detection, after applying the erosion operation, a dilation operation was performed based on structural elements with the size of 30×30 pixels. The size was determined based on the image resolution and the criterion of filling small holes in clouds that appeared after applying the erosion operation. In the case of detection of the sun, the application of the dilation operation was not necessary since it is the most glittering element on the image, only erosion. The results of the transformation can be seen in Figure 7d.

After the sun and cloud contours have been detected and saved, altogether with their centroid points, the next phase of our algorithm is to determine the sun coverage level based on cloud's and the sun's speed vectors, described in the following section.

4. Determining the Sun Coverage Level

4.1. Determining Candidate Clouds for Sun Coverage

The detected cloud boundaries with their centroid points and minimum-area bounding rectangle are shown in Figure 8a. The cloud boundaries are marked in blue, the centroids in red, and the minimum area bounding rectangles in green. The origin of the coordinate system (0, 0) is located in the upper left corner of the image.

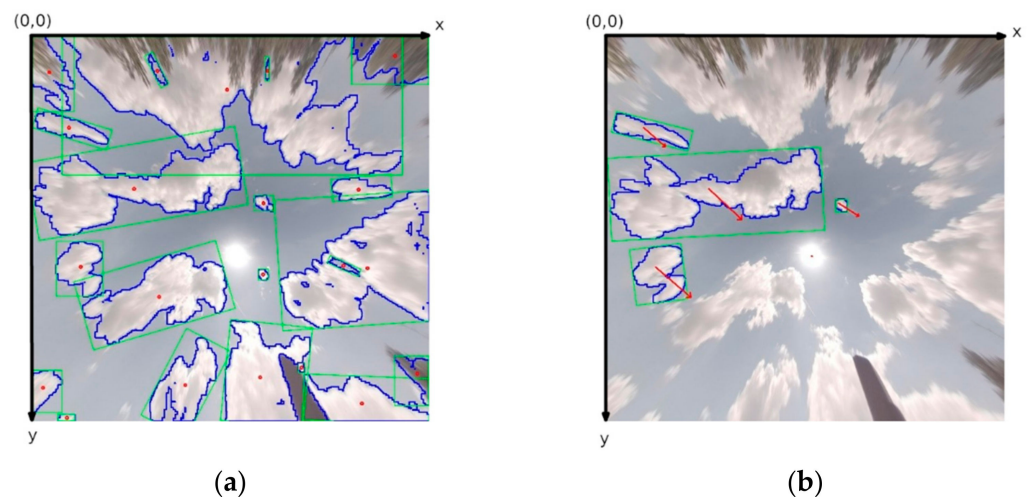


Figure 8. Cloud centroid position change within 5 min: (a) detected clouds, (b) candidate clouds for sun coverage.

Although the sun is also one of the contours detected, it is not selected because the centroid of the sun's contour is the same centroid already detected in the previous step of detecting only the sun.

Figure 8b shows the direction of cloud movement based on the location change of the centroid points. The red arrow represents the direction and location change within 5 min. Therefore, the length of the red arrows represents the movement speed of the clouds.

Candidate clouds are determined by the current position of their center of gravity in relation to the position of the sun's center of gravity. If the clouds are behind the sun (and can no longer cover the sun) or are too far from the sun (the simulated motion in the future

never covers the sun and does not come close to the sun), they are not tracked. Figure 8b shows a situation where only four clouds were tracked because they met the requirements for tracking.

The origin with coordinates (0, 0) is located in the upper left corner of the image, and thus the contour and rectangular shape approximation of the sun is simulated with an arrow, shown in Figure 9.



Figure 9. Change of the sun's position within 1 h.

4.2. Determination of the Velocity Vectors of Clouds and the Sun

A sequence of 12 images was used to determine the velocity vectors needed to estimate cloud and sun motion. The time required for the ground-based camera to capture 12 images is 60 s (12×5 s). The position change is calculated based on the motion during the last minute. Figure 10 shows a flowchart describing the sun coverage detection algorithm that was used. Within the Supplementary Material to this paper, there is a source code in Java that implements described algorithm. Since the position of the sun and the position of all clouds were determined in the previous part of the algorithm, the velocity vector is determined based on the displacement of the cloud centroids within 60 s. In finding the nearest centroid position after Δt (60 s), the nearest centroid was used, and the equation for the distance between two points in a coordinate system was used to find the nearest centroid of all clouds detected on the sky image $d(c_2, c_1)$:

$$d(c_2, c_1) = \sqrt{(x_{c_2} - x_{c_1})^2 + (y_{c_2} - y_{c_1})^2} \quad (7)$$

$$\vec{v}_i = (x_{c_2} - x_{c_1}) \vec{i} + (y_{c_2} - y_{c_1}) \vec{j}$$

where c_2 and c_1 represent two nearest centroids, x_{c_1} and x_{c_2} are x coordinates of the centroids, y_{c_1} and y_{c_2} are y coordinates of the centroids, and \vec{v}_i is the speed vector. The speed vector \vec{v}_i was determined by using the x and y axis shifts of the centroid points after Δt and by multiplying it with unit vectors \vec{i} and \vec{j} .

The flow diagram in Figure 10 starts with initiating time of 0, selecting the first cloud with the index i , and setting the counter k to 0.

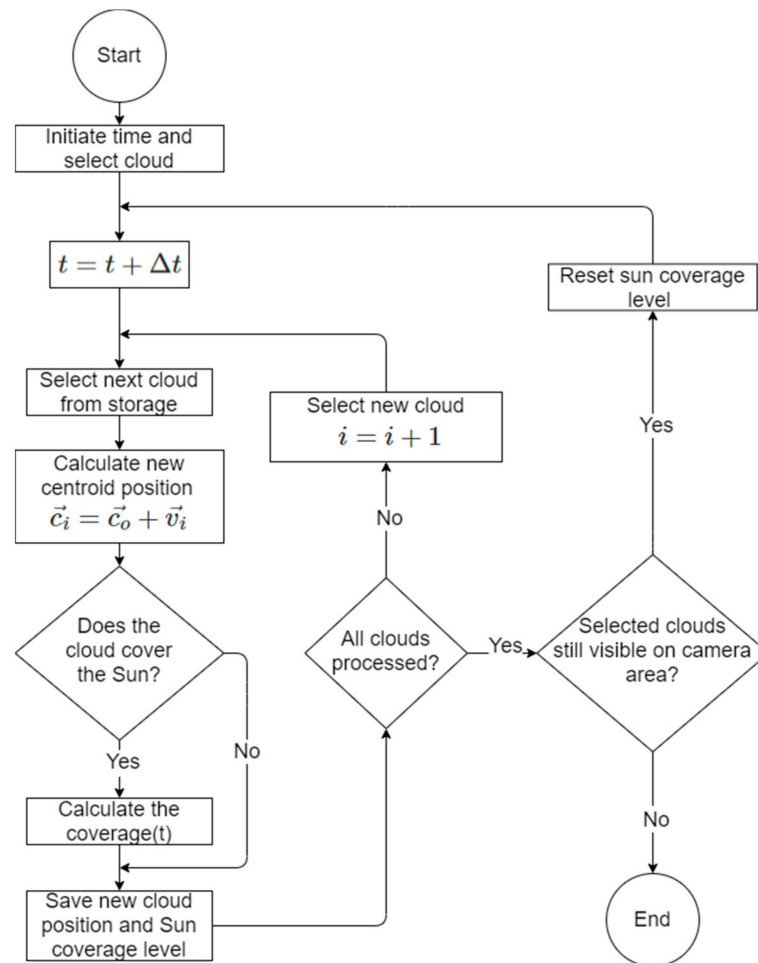


Figure 10. Sun coverage estimation algorithm flow diagram.

The outer loop increments the counter, time $t(k)$ by Δt , and calculates the new centroid position of the cloud with the index i :

$$\begin{aligned} \vec{c}_{i,k+1} &= \vec{c}_{i,k} + \vec{v}_{i,k}\Delta t \\ t(k) &= t(k-1) + \Delta t \end{aligned} \quad (8)$$

where $\vec{c}_{i,k}$ represents the motion vector of cloud i in the moment defined with counter k , $\vec{c}_{i,k+1}$ represents the motion vector of the cloud i in the moment defined in the next period (5 s) $k + 1$, $\vec{v}_{i,k}$ represents the speed vector, Δt is the time period (5 s), $t(k)$ is the time after k periods, and $t(k - 1)$ is the time in the previous period.

The goal of the algorithm is to determine the time signal of the sun's degree of coverage for the next hour. This signal is defined by two fields, t and coverage, which together represent a discrete form of the degree of coverage over time. For this purpose, two iterative processes have been established: the inner loop, which calculates the solar occultation at a given instant by estimating the position of all the clouds for that instant and calculating the total solar coverage, and the outer loop, which determines the change in time to obtain the position and the solar coverage for all instants, i.e., to calculate the signal coverage (t).

The procedure for determining the total curvature of the sun at a given time in the inner loop consists of several steps. In the first step, the first cloud suitable for occulting the sun is selected, and a new position is calculated using its old position and velocity

according to Equation (8). For the new position of the sun and the cloud, the degree of coverage of the sun is calculated, and the new cloud and sun position and the degree of coverage are stored for that cloud and time. Then, the next cloud is selected to estimate the degree of occultation. After the analysis of all clouds is completed, the final solar coverage is determined as the sum of the coverages of all selected clouds for the respective time.

Figure 11 shows the simulated motion of clouds with different sizes and speeds approximated by a rectangular shape every 5 min.

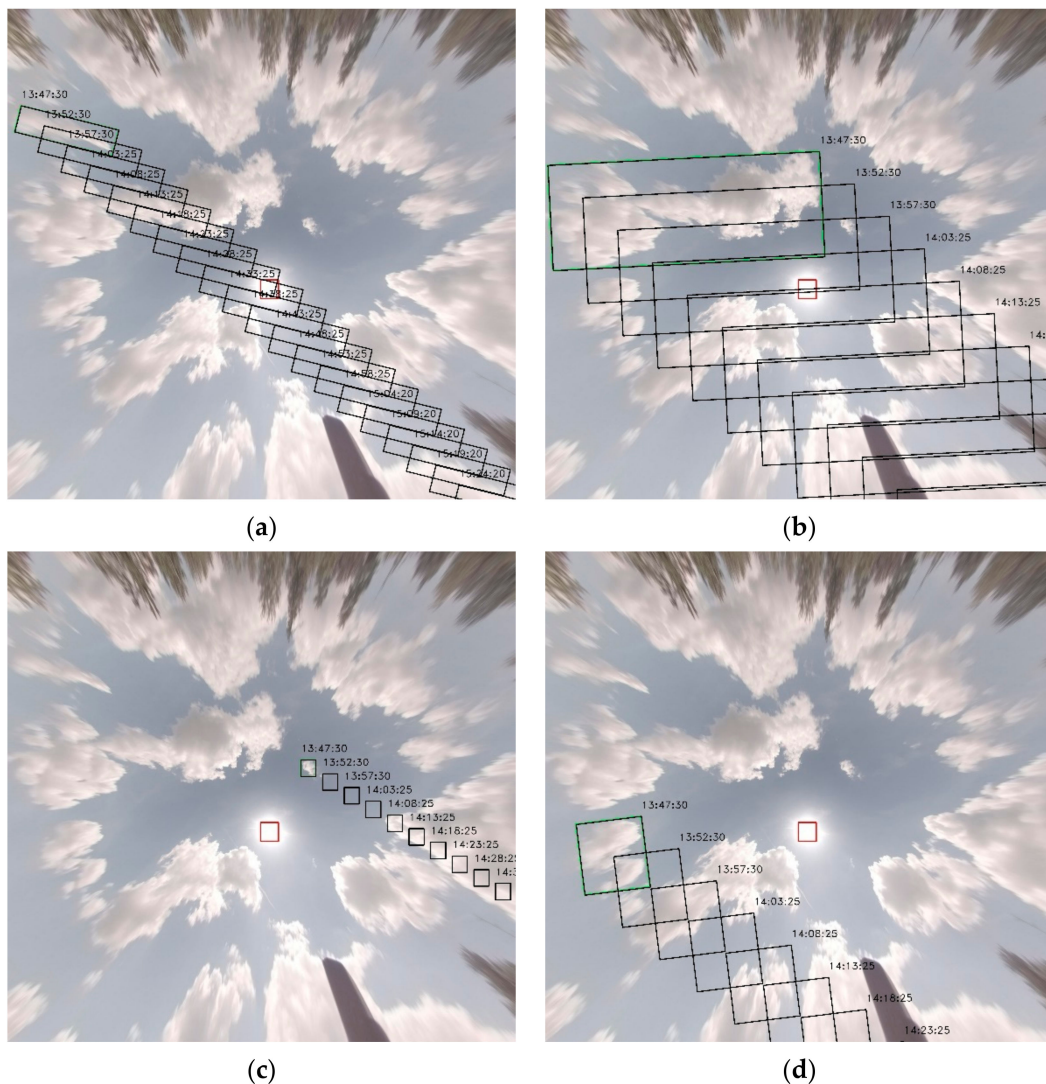


Figure 11. Estimated positions of the clouds based on the movement simulation: (a) for the cloud that entered the image and has the lowest speed, (b) for the biggest cloud that is tracked, (c) for the smallest cloud that is tracked, (d) for the cloud with the highest speed.

For each of the simulated positions of the cloud present in the scene, the degree of solar coverage is determined. The collected parameters were used to simulate the movement of the clouds in the following period. The contours of the clouds and the sun were approximated by minimum-area bounding rectangles. Each cloud rectangle is marked with a timestamp representing the position of the cloud at that time. The intersection between the clouds and the sun is determined using the “rotatedRectangleIntersection” method from the “Imgproc” class. If the clouds cover the entire sun, then the coverage is set to 100% (the total area of the sun in pixels is the result of the method call); otherwise, it is less than 100%.

5. Results

After the cloud motion simulation was performed, the degree of sun coverage was calculated for each time period in the future as long as the tracked clouds were present in the undistorted image. The comparison of the actual and predicted sun coverage level is shown in Figures 12 and 13. The data shown covers the period on 13 July 2020, from 13:45 to 14:40 and from 12:00 to 12:45.

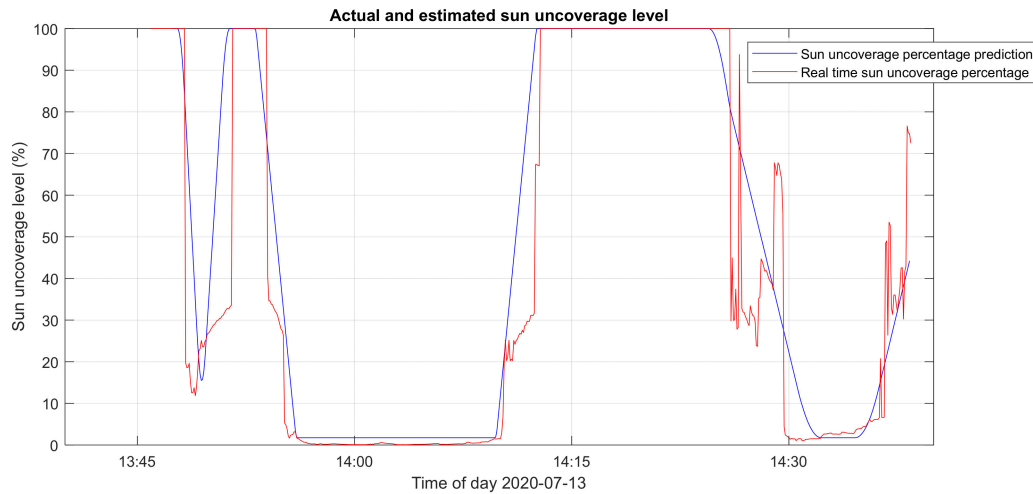


Figure 12. Comparison of the real-time sun un-coverage and the predicted level of sun un-coverage on 13 July 2020 from 13:45 to 14:40.

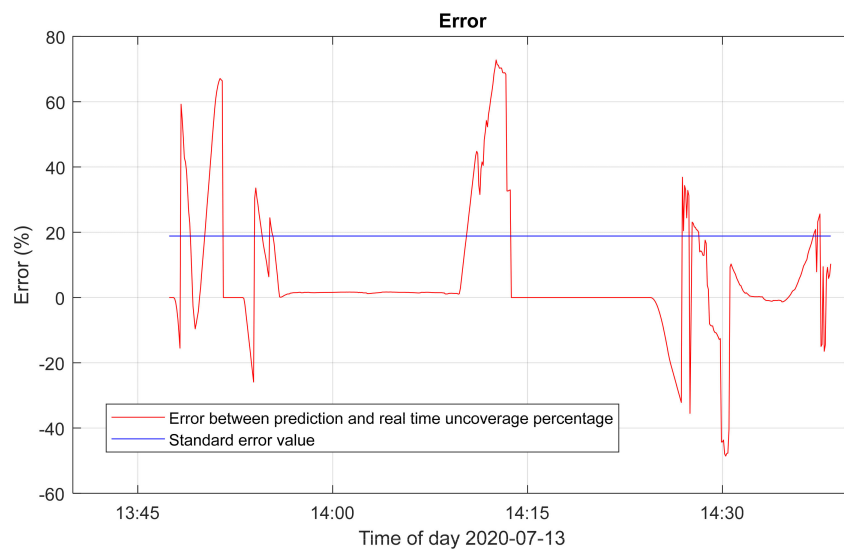


Figure 13. Error level between real-time and predicted level of sun un-coverage on 13 July 2020 from 13:45 to 14:40.

Figures 12–19 show the situation in the sky where several clouds were moving towards the sun and corresponding error levels. The red curve, which represents the real-time sun un-coverage level on images that compare the real-time and predicted sun-uncoverage level, shows that the visible area of the sun changes significantly in a nonlinear manner as the clouds begin to cover or uncover the sun. This effect is caused by the transparent nature of the clouds, which prevents the sun from being completely covered at the beginning of the cloud coverage process, while the rectangular shape approximation does not take this effect into account. The real-time sun un-coverage level is based on the detected area of the sun when it is covered with clouds compared to the area of the sun when there are no clouds. The corresponding error levels are marked with red curve, and standard error value with blue curve.

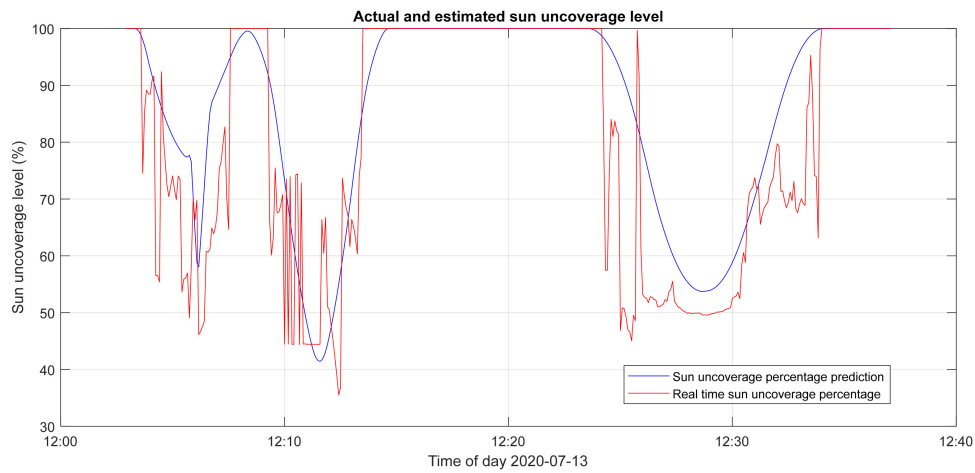


Figure 14. Comparison of real-time sun un-coverage and predicted level of sun un-coverage on 13 July 2020 from 12:00 until 12:40.



Figure 15. Error level between real-time and predicted level of sun un-coverage on 13 July 2020 from 12:00 until 12:40.

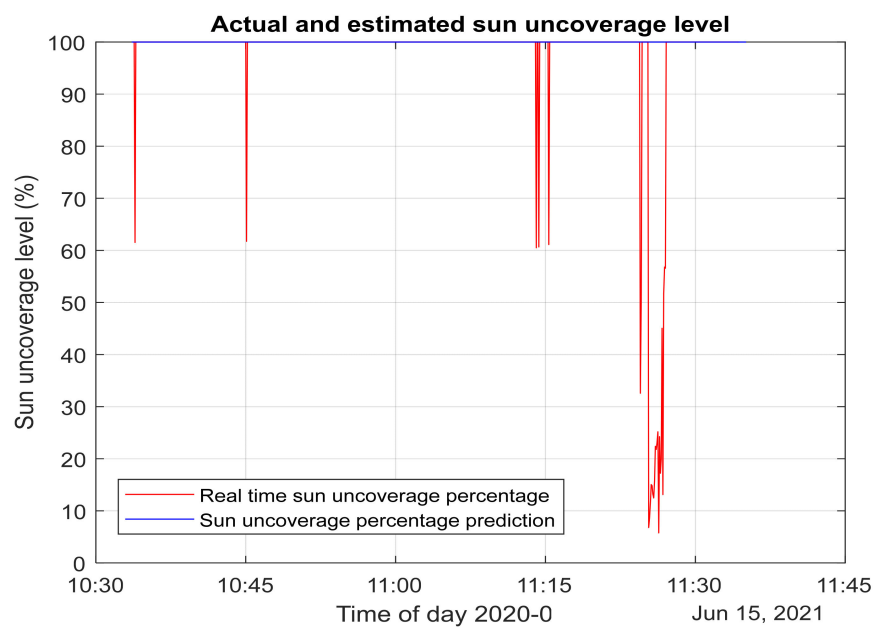


Figure 16. Comparison of real-time sun un-coverage and predicted level of sun un-coverage on 14 July 2020 from 10:30 until 11:45.

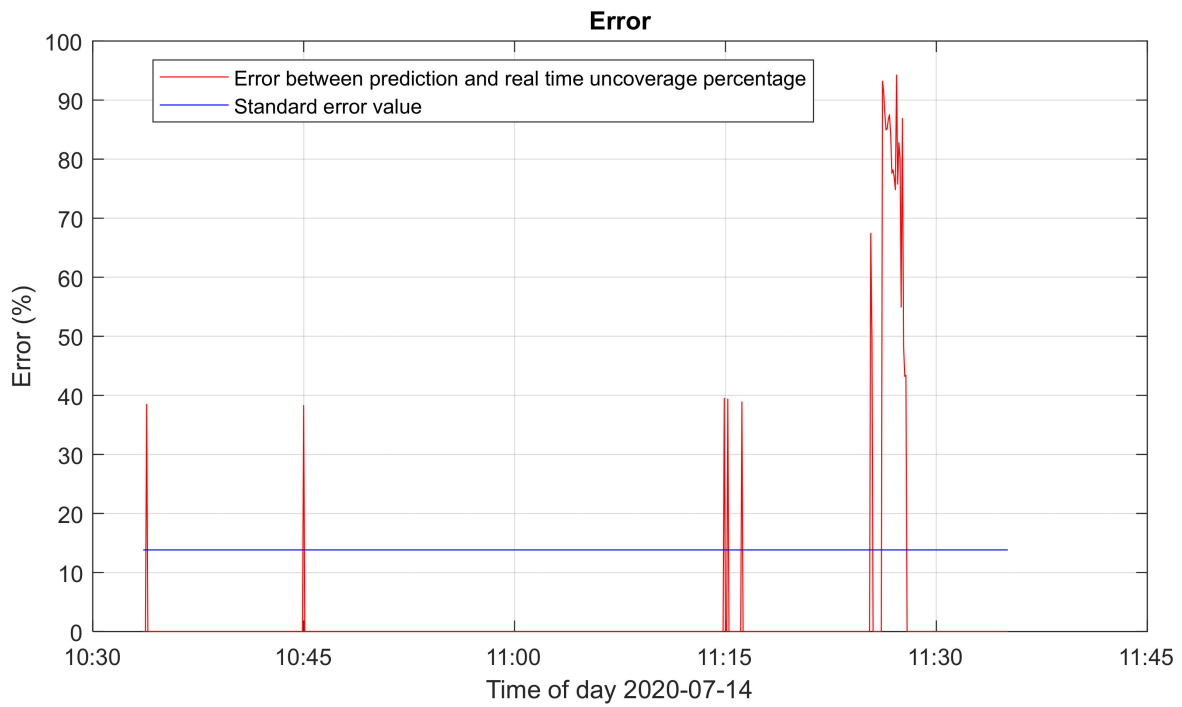


Figure 17. Error level between real-time and predicted level of sun un-coverage on 14 July 2020 from 10:30 until 11:45.

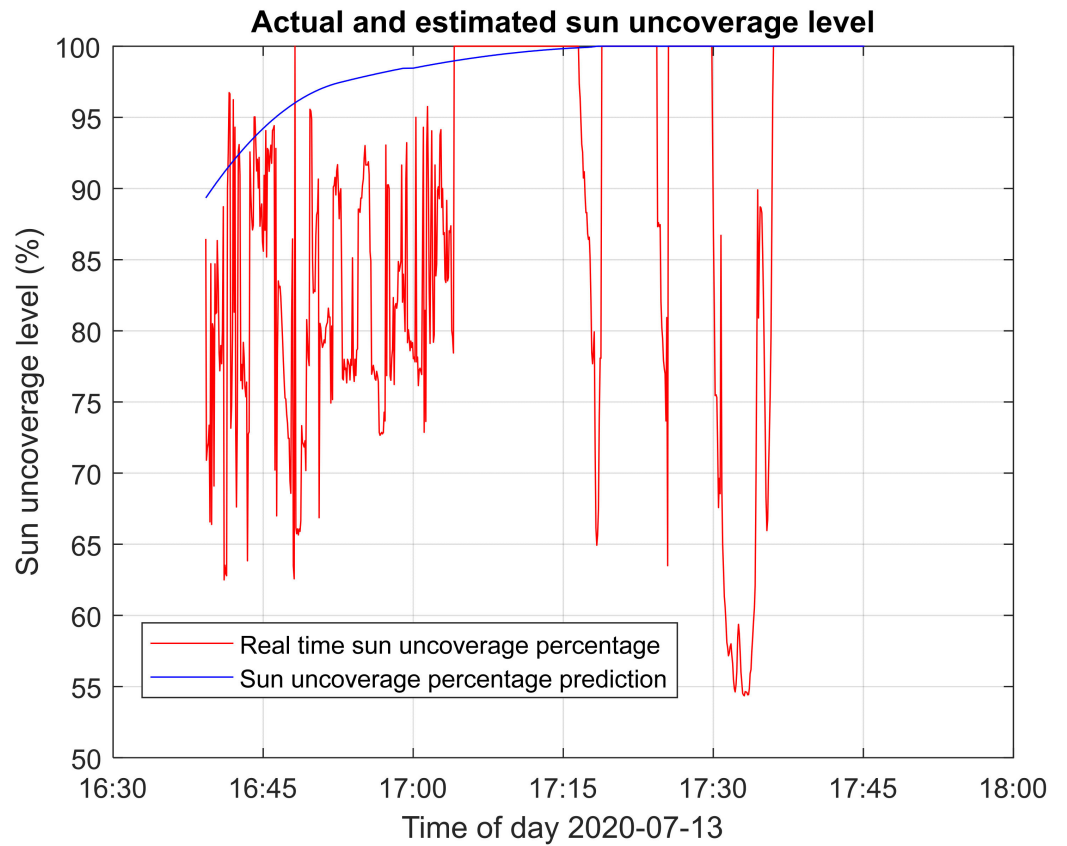


Figure 18. Comparison of real-time sun un-coverage and predicted level of sun un-coverage on 13 July 2020 from 16:30 until 17:45.



Figure 19. Error level between real-time and predicted level of sun un-coverage on 13 July 2020 from 16:30 until 17:45.

The predicted level of coverage, based on the rectangular shape approximation obtained by the described algorithm, covers and uncovers the sun region in a linear manner, as shown by the blue curve in Figures 12, 14, 16 and 18, with corresponding prediction error levels shown on Figures 13, 15, 17 and 19, respectively. The blue curve in Figures 12, 14, 16 and 18 represents a prediction. It was performed at the beginning of the period of 1 h: in Figure 12 at 13:47, in Figure 14 at 12:03, in Figure 16 at 10:34, and in Figure 18 at 16:38. After the algorithm analyzed the position change of the clouds and the sun during the last minute (12 images were used, each image was taken every 5 s), the prediction was performed for the next hour. The prediction (blue curve) was compared to the real-time un-coverage level (red curve) that was measured after the prediction was performed to calculate the error level. The real-time curves and the predictions match in the critical areas where the sun is fully covered or fully exposed.

The obtained curve can be used to predict the amount of direct solar irradiance calculated according to the following equation:

$$EI(t) = PI(t) \frac{PAI(1h)_{CMHS}}{PSUL(1h)_{Calculated}} \quad (9)$$

where $EI(t)$ represents estimated irradiance in $\frac{W}{m^2}$, $PI(t)$ is the estimated sun un-coverage level measured in percentages (represented by the blue curves in Figures 12, 14, 16 and 18), $PAI(1h)_{MHS}$ represents predicted average irradiance estimation of direct solar irradiance based on the 1-h time period provided by the meteorological and hydrological service measured in $\frac{W}{m^2}$, and $PSUL(1h)_{Calculated}$ represents the predicted average value of predicted sun un-coverage level (measured in percentage from 0 to 100%).

The standard error level shown in Figures 13, 15, 17 and 19 shows that it is in range of 10–20%. The results show that it is the highest in periods when clouds start and with sun coverage, especially if the clouds are transparent and the sunlight can break through them.

In cases when the sun is positioned near to the horizon, with an orientation where the clouds come from to the horizon (for example, the east side is on the left part of the taken sky images) during the morning (as shown in Figure 16), the prediction possibilities are limited. This is because of the short distance between the sun and the clouds. However, total irradiance is usually lower during the morning, so the absolute error related to the energy amount is lower as well. Therefore, optimal prediction is obtained when the sun is positioned further away from the cloud positions (for example, in the middle of the day, during noon).

The meteorological and hydrological service provides forecast values for direct solar irradiance for a selected area for the next 64 h and provides an average value of solar

irradiance per hour. Based on the prediction of the sun coverage with clouds, the average value of direct solar irradiance can be estimated, as shown in Figures 20–23.

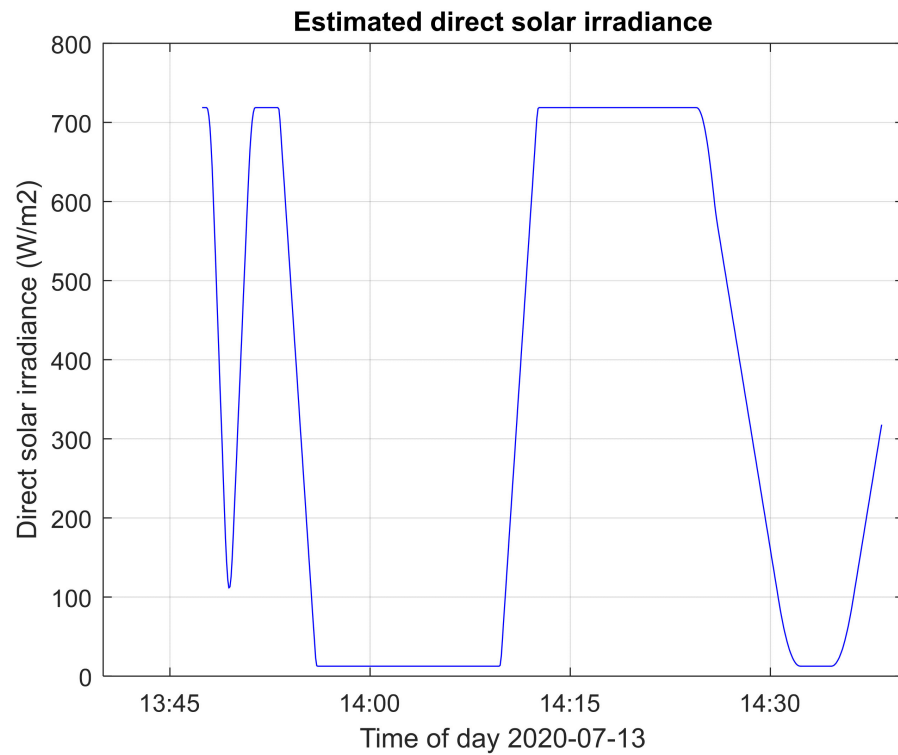


Figure 20. Predicted level of direct solar irradiance on 13 July 2020 from 13:45 until 14:40.

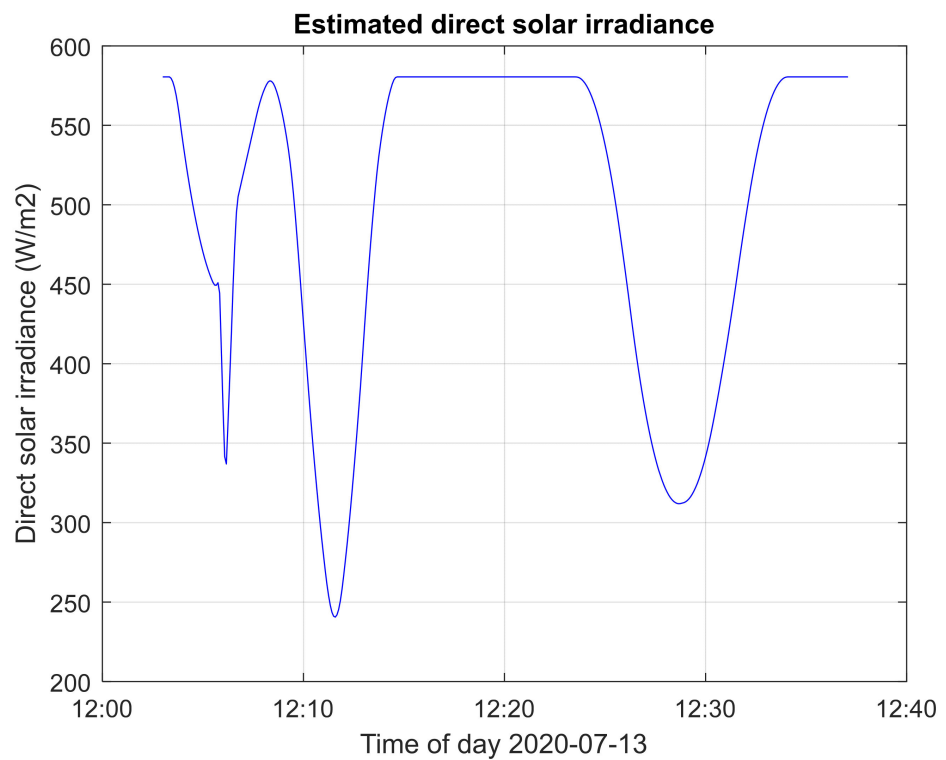


Figure 21. Predicted level of direct solar irradiance on 13 July 2020 from 12:00 until 12:40.

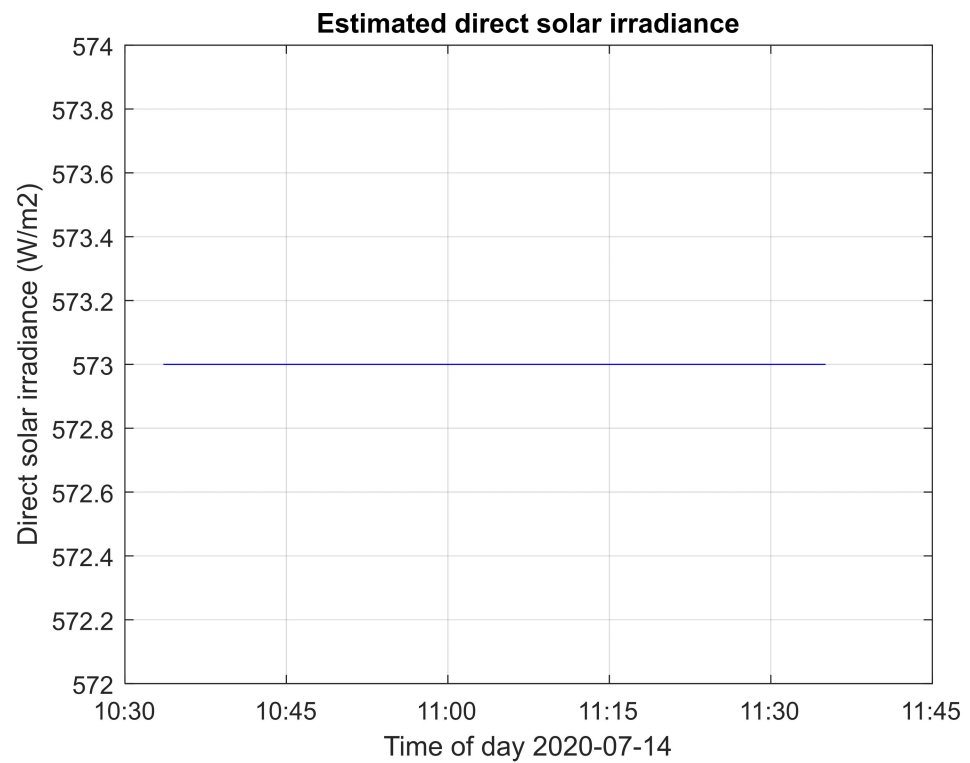


Figure 22. Predicted level of direct solar irradiance on 14 July 2020 from 10:30 until 11:45.

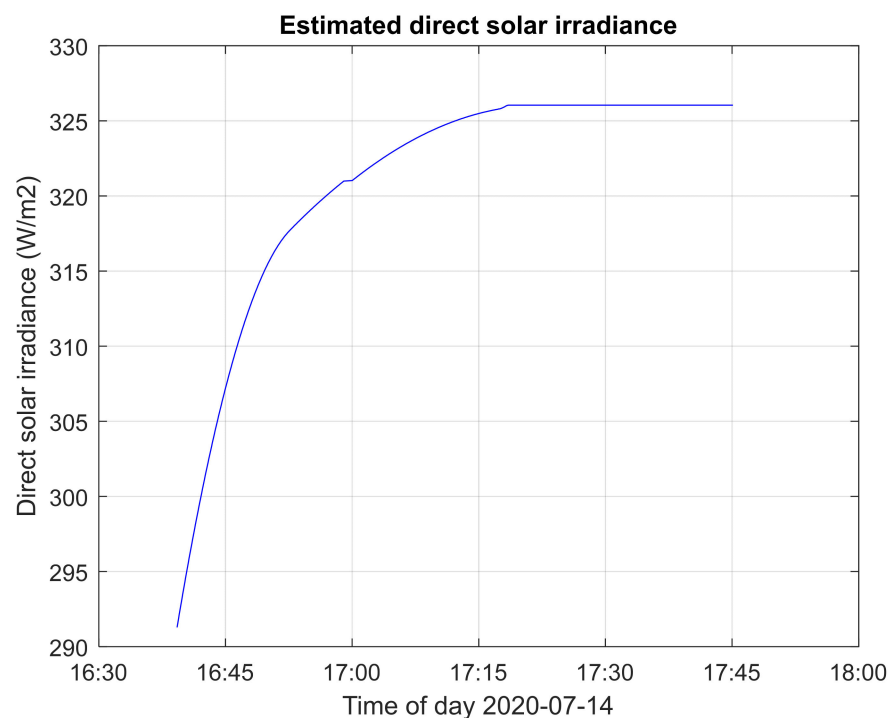


Figure 23. Predicted level of direct solar irradiance on 13 July 2020 from 16:40 until 17:45.

Since the minimum-area bounding rectangle shape approximations were used for the cloud and solar contours, the algorithm proved to be very simple and requires an average time for estimating the sun coverage between 5 and 6 s. The simulations were run on a personal computer with the Intel Core i7-8750 processor (2.20 GHz) with 24 GB of RAM (1 GB was used for the integrated development environment for Java) with an SSD disk drive with sequential reading up to 2800 MB/s.

6. Conclusions

The results show that the cloud positions can be determined with respect to the position of the sun if the contour shapes are approximated with a minimum-area bounding rectangle. The shape approximation has no significant effect on the estimation of the beginning and end of the cloud shadow over the sun. The simplicity and speed of the described algorithm make our approach for estimating direct solar irradiance suitable for use in solar-based virtual power plants. Error measured between the predicted and real-time un-coverage can be used as a security limit when designing the storage systems. The predicted amount of generated energy can be decreased to avoid wrong optimistic predictions that can affect the stability of a virtual power plant. The error cannot be determined online, but based on experience gained through the measurements and different periods during the day.

The presented results show that a short-term prediction based on a ground camera and the use of the estimated degree of sun coverage in combination with the hourly average solar irradiance forecast provided by the meteorological and hydrological service can be used to predict the amount of direct solar irradiance. The short-term prediction is performed as soon as a cloud appears on the sky image captured by the camera so that after a fixed period of time has elapsed (1 min or 12 images captured every 5 s), the estimation process can be automatically started. The only limitation is based on solar visibility. The estimation process can begin when the initial position of the sun has been captured.

Future work will include comparison with measurement data collected with a pyranometer instrument and will include all periods of the year, including fall, winter, spring, and summer, to cover all possible types of clouds. The options, such as adapting the system to different seasons and cloud types, can be upgraded with a machine learning component that replaces manual human work and improves the autonomy of the system. In addition, the direct solar irradiation curve obtained by prediction could be used to optimize the energy flows in a microgrid with a photovoltaic system, including energy storage devices.

The article shows one of the methods for short-term solar irradiance prediction based on cloud motion prediction. The study involved the installation of a manual camera placed on the roof of the residential building and offline processing of the images. The results prove that a microgrid system based on different energy storage devices with different parameters (such as the capacity, the delay between the change of mode from charging to discharging, etc.) is suitable for short-term prediction, focusing on the period from half an hour to 1 h.

Supplementary Materials: The following are available online at <https://www.mdpi.com/article/10.3390/en14133775/s1>.

Author Contributions: Conceptualization, A.R. and Ž.B.; methodology, A.R. and Ž.B.; software, A.R. and D.K.; validation, Ž.B., D.K., and V.Š.; formal analysis, Ž.B.; investigation, A.R. and Ž.B.; resources, Ž.B.; data curation, A.R. and Ž.B.; writing—original draft preparation, A.R.; writing—review and editing, A.R.; visualization, A.R.; supervision, Ž.B. and V.Š.; project administration, Ž.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, T.; Guo, Y.; Wang, C.; Ni, C. Inter-Hour Forecast of Solar Radiation Based on the Structural Equation Model and Ensemble Model. *Energies* **2020**, *13*, 4534. [[CrossRef](#)]
2. Pasion, C.; Wagner, T.; Koschnick, C.; Schuldt, S.; Williams, J.; Hallinan, K. Machine Learning Modeling of Horizontal Photovoltaics Using Weather and Location Data. *Energies* **2020**, *13*, 2570. [[CrossRef](#)]

3. Wang, J.; Qian, Z.; Wang, J.; Pei, Y. Hour-Ahead Photovoltaic Power Forecasting Using an Analog Plus Neural Network Ensemble Method. *Energies* **2020**, *13*, 3259. [[CrossRef](#)]
4. Wojtkiewicz, J.; Hosseini, M.; Gottumukkala, R.; Chambers, T.L. Hour-Ahead Solar Irradiance Forecasting Using Multivariate Gated Recurrent Units. *Energies* **2019**, *12*, 4055. [[CrossRef](#)]
5. Aslam, M.; Lee, J.-M.; Kim, H.-S.; Lee, S.-J.; Hong, S. Deep Learning Models for Long-Term Solar Radiation Forecasting Considering Microgrid Installation: A Comparative Study. *Energies* **2019**, *13*, 147. [[CrossRef](#)]
6. Andrianakos, G.; Tsourounis, D.; Oikonomou, S.; Kastaniotis, D.; Economou, G.; Kazantzidis, A. Sky Image forecasting with Generative Adversarial Networks for cloud coverage prediction. In Proceedings of the 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Greece, 15–17 July 2019; pp. 1–7. [[CrossRef](#)]
7. Maitanova, N.; Telle, J.-S.; Hanke, B.; Grottko, M.; Schmidt, T.; Von Maydell, K.; Agert, C. A Machine Learning Approach to Low-Cost Photovoltaic Power Prediction Based on Publicly Available Weather Reports. *Energies* **2020**, *13*, 735. [[CrossRef](#)]
8. Kwon, Y.; Kwasinski, A.; Kwasinski, A. Solar Irradiance Forecast Using Naïve Bayes Classifier Based on Publicly Available Weather Forecasting Variables. *Energies* **2019**, *12*, 1529. [[CrossRef](#)]
9. Lešić, V.; Martinčević, A.; Vašak, M. Modular energy cost optimization for buildings with integrated microgrid. *Appl. Energy* **2017**, *197*, 14–28. [[CrossRef](#)]
10. Miñambres-Marcos, V.M.; Guerrero-Martínez, M.Á.; Barrero-González, F.; Milanés-Montero, M.I. A Grid Connected Photovoltaic Inverter with Battery-Supercapacitor Hybrid Energy Storage. *Sensors* **2017**, *17*, 1856. [[CrossRef](#)]
11. Yu, A.; Chabot, V.; Zhang, J. *Electrochemical Supercapacitors for Energy Storage and Delivery: Fundamentals and Applications*; Taylor & Francis: New York, NY, USA, 2017; pp. 1–355. [[CrossRef](#)]
12. Miyazaki, Y.; Kameda, Y.; Kondoh, J. A Power-Forecasting Method for Geographically Distributed PV Power Systems using Their Previous Datasets. *Energies* **2019**, *12*, 4815. [[CrossRef](#)]
13. Kazantzidis, A.; Tzoumanikas, P.; Blanc, P.; Massip, P.; Wilbert, S.; Ramirez-Santigosa, L. Short-term forecasting based on all-sky cameras. In *Renewable Energy Forecasting*; Woodhead Publishing: Cambridge, UK, 2017; pp. 153–178. [[CrossRef](#)]
14. Li, X.; Lu, Z.; Zhou, Q.; Xu, Z. A Cloud Detection Algorithm with Reduction of Sunlight Interference in Ground-Based Sky Images. *Atmosphere* **2019**, *10*, 640. [[CrossRef](#)]
15. Radovan, A.; Ban, Z. Prediction of HSV color model parameter values of cloud movement picture based on artificial neural networks. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 1110–1114. [[CrossRef](#)]
16. Dev, S.; Lee, Y.H.; Winkler, S. Color-Based Segmentation of Sky/Cloud Images From Ground-Based Cameras. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2017**, *10*, 231–242. [[CrossRef](#)]
17. Tuominen, P.; Tuononen, M. Cloud detection and movement estimation based on sky camera images using neural networks and the Lucas-Kanade method. *AIP Conf. Proc.* **2017**, *1850*. [[CrossRef](#)]
18. Kamadinata, J.O.; Ken, T.L.; Suwa, T. Global Solar Radiation Prediction Methodology using Artificial Neural Networks for Photovoltaic Power Generation Systems. In Proceedings of the SMARTGREENS 2017—6th International Conference on Smart Cities and Green ICT Systems, Porto, Portugal, 22–24 April 2017; pp. 15–22. [[CrossRef](#)]
19. Sukič, P.; Štumberger, G. Intra-Minute Cloud Passing Forecasting Based on a Low Cost IoT Sensor—A Solution for Smoothing the Output Power of PV Power Plants. *Sensors* **2017**, *17*, 1116. [[CrossRef](#)] [[PubMed](#)]
20. Jang, H.S.; Bae, K.Y.; Park, H.-S.; Sung, D.K. Solar Power Prediction Based on Satellite Images and Support Vector Machine. *IEEE Trans. Sustain. Energy* **2016**, *7*, 1255–1263. [[CrossRef](#)]
21. Chauvin, R.; Nou, J.; Thil, S.; Grieu, S. Cloud motion estimation using a sky imager. *AIP Conf. Proc.* **2016**, *1734*. [[CrossRef](#)]
22. Kaczorowska, D.; Rezmer, J.; Jasinski, M.; Sikorski, T.; Suresh, V.; Leonowicz, Z.; Kostyla, P.; Szymanda, J.; Janik, P. A Case Study on Battery Energy Storage System in a Virtual Power Plant: Defining Charging and Discharging Characteristics. *Energies* **2020**, *13*, 6670. [[CrossRef](#)]
23. Wang, H.; Kong, H.; Pu, Z.; Li, Y.; Hu, X. Feasibility of high efficient solar hydrogen generation system integrating photovoltaic cell/ photon-enhanced thermionic emission and high-temperature electrolysis cell. *Energy Convers. Manag.* **2020**, *210*, 112699. [[CrossRef](#)]
24. Avila, E.; Pozo, M.; Pozo, N.; Camacho, O.; Leica, P.; Ortega, L.; Gallardo, C.; Dominguez, X.; Feuersänger, S.; Pacas, M. Energy management of a virtual power plant with a battery-ultracapacitor based Hybrid Energy Storage System. In Proceedings of the 2017 IEEE Southern Power Electronics Conference (SPEC), Puerto Varas, Chile, 4–7 December 2017; pp. 1–6. [[CrossRef](#)]
25. Abdolrasol, M.G.M.; Mohamed, A.; Hannan, M.A. Virtual power plant and microgrids controller for energy management based on optimization techniques. *J. Electr. Syst.* **2017**, *13*, 285–294.
26. Ruffli, M.; Scaramuzza, D.; Siegwart, R. Automatic detection of checkerboards on blurred and distorted images. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3121–3126. [[CrossRef](#)]
27. Scaramuzza, D.; Martinelli, A.; Siegwart, R. A Toolbox for Easily Calibrating Omnidirectional Cameras. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006. [[CrossRef](#)]
28. Scaramuzza, D.; Martinelli, A.; Siegwart, R. A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS'06), New York, NY, USA, 4–7 January 2006.

-
29. Matlab Function 'showExtrinsics'. Available online: <https://www.mathworks.com/help/vision/ref/showextrinsics.html> (accessed on 11 January 2021).
 30. Radovan, A.; Ban, Ž. Predictions of cloud movements and the sun cover duration. In Proceedings of the 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 26–30 May 2014; pp. 1210–1215. [[CrossRef](#)]
 31. OpenCV Structural Analysis and Shape Descriptors. Available online: https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html (accessed on 1 May 2021).