EMERGING TECH CONFERENCE – Edge Intelligence

Volume 02, 2023, Page 8 – 15

Proceedings of Emerging Tech Conference:
Edge Intelligence 2023

Comparing the Performance of Cameras and SW packets for the Cloud Coverage Process in Photovoltaic (PV) Parks

C. Vasilakis[a], G. Venitourakis[a], A. Tsagkaropoulos[a], P. Tz. Amrou[b], G. Konstantoulakis[b], P. Golemis[b], D. Reisis[a]*

[a] *Electronics Lab, Physics Dpt, National & Kapodistrian University of Athens, Greece*
[b] *InAccess Networks, 12 Sorou Str, 15125, Athens, Greece.*
** Corresponding author. Email address: dreisis@phys.uoa.gr*

Abstract

The clouds' position in the sky, their creation, movements and dissipation have a significant impact on the energy produced in photovoltaic (PV) parks. To improve the power supply and the grid balance, the sky cameras will provide the images of the sky and edge computing devices will estimate the clouds' coverage in the sky and consequently compute the upper amount of energy that can be produced. For the execution of these processes the PV plants depend on systems of considerable cost that include the cameras and the supporting software most often executed on edge computing devices. The requirements for extreme weather conditions of these application specific cameras and their supporting hardware and software lead to high-cost figures. These facts decrease the efficiency defined as the results accuracy over the system's cost. Aiming at efficient solutions in the course of developing a PV park controller [1], the current study presents and compares the results of applying a cloud coverage process on systems with cameras of different cost and specifications. Moreover, it proposes an effective image segmentation technique for the cloud coverage executed on edge devices and it compares its accuracy to that of commercially available software. The comparison analyzes the differences in the accuracy results of three system configurations using each of two different cameras and either the proposed cloud segmentation algorithm or the commercial software. The real time performance of the proposed system algorithmic techniques on the edge-oriented NVIDIA Jetson Nano [2] validates the techniques.

Keywords: Image Segmentation, Cloud Coverage, Edge Computing

1 Introduction

The photovoltaic (PV) power plants have been integrated in the power grid and they continuously enhance their contribution to the energy produced. Their power production strongly depends on the cloud variability and on the seasonal conditions of their geographical location. How the clouds are created, how the parks are obscured by the clouds and how they disappear, has a significant effect on the power production especially in the large-scale power plants. Therefore, the estimation of the PV park's power production by the PV park controller in the time window of 5 minutes is crucial for improving the production as well as for the grid balance. This estimation becomes necessary in order to prevent the abrupt loss of a major fraction of the electricity production and possible blackouts in the park's neighboring

industrial, business and residential areas. To calculate the curve of the physical limits of the PV power, the park controller has to estimate the cloud coverage based on the current sky images. For this purpose, it includes a system based on a camera featuring an 180° "fisheye" lens to take all-sky pictures. These cameras are called sky imagers and since they are placed in the open most often, they have specifications for extreme weather conditions. Such complete systems are commercially available at a relatively high cost (in the class of 10-30 K€). Moreover, such systems' vendors provide software for image segmentation more often (specifically for cloud coverage). For a large number of smaller size parks though, the weather conditions are mild over the entire year and so, the high cost of such a complete system including a sky imager and the software becomes a challenge.

Targeting a cost-efficient solution to the cloud coverage problem the current paper shows the results of the research focusing on improving the cloud coverage estimation while employing systems of reduced cost, especially in geographical locations and places where these installations are feasible. The most significant contribution of this work is an image segmentation algorithm for cloud coverage, which can be executed on edge computing devices, and it was developed in the Archon project [1]. The proposed algorithm utilizes three techniques for: a) locating the sun in the sky image, b) removing from the image all the objects that irrelevant to the clouds and they are located close to the image's margin, such as buildings, antennas, etc. and c) classifications of all the pixels as cloudy or cloudless. The results of the comparison of the performance between the proposed algorithm and commercially available software [3] are equally important. The cameras used in the comparison are [3] and a low-cost camera equipped with fisheye lens [4]. The comparison involves three configurations of a complete system. The first consists of the sky imager ASI-16/52 system and this system's commercial software computing the cloud coverage [3]. The second includes the sky imager ASI-16/52 and the proposed image segmentation technique for cloud coverage. The third system uses the low-cost camera Vivotek [4] with the proposed Archon image segmentation software. The motivation for this work was the cost improvement of the PV park controller systems while keeping the performance with respect to the accuracy of the results and achieve the real time execution requirements on an edge computing device.

It is worth mentioning that reducing the cost of the cameras and their supporting hardware is realistic only in geographical locations of temperate climate and mild weather conditions. For a complete study and comparison there were tested various configurations, with each configuration consisting of a mixture of commercially available hardware and software solutions as well as software techniques developed in the project. This paper reports the three noteworthy combinations' results.

2 Related Work

There is a plethora of research results related to cloud coverage solutions, which target the accuracy improvement. Most of these reports consider satellite images while the number of studies in the category of techniques using images taken from the ground constantly increases [5]. For the latter category, the first process in the flow is locating the sun in a sky image, which is essential for cloud segmentation algorithms [6]. The authors of [6] introduce a sun tracking algorithm, where the sun's trajectory is estimated by correlating the position of the sun in a sky image with the timestamp of the image for a set of clear sky images with short exposure. The sun's location is revealed by thresholding clear sky images with short exposure based on pixel intensity. This method is camera and location agnostic but requires a dataset in order to identify the sun's trajectory. Simultaneously, results from [7] suggest that sun localization data improve the Forecast Skill (FS) of several CNN (Convolutional Neural Network) models up

to 13.75% for the task of irradiance forecasting. The second process isolates and removes from the image the ground objects [8] with mostly straightforward techniques such as removing the pixels close to the image margin. An alternative - automated - way to remove the obstacles and the noise from the images is the Machine Learning (ML) approach proposed in [9]. The authors use supervised machine learning for context awareness in addition to PCA (Principal Component Analysis) and HCA (Hierarchical Cluster Analysis) to reduce the data dimensionality. Focusing on clouds in images [10] and plant management, D. K Danso et al. conclude that ground cameras are more effective than satellite imagery. In [11], a multi-color criterion is proposed for the estimation of total cloud coverage of a whole-sky image system (commercial camera) by classifying the digital images in seven cloud classes and achieving 83% and 94% accuracy. The [12] study proposes a modification of the cloud detection Red-Blue Ratio (RBR) threshold by incorporating the difference between the two channels instead of their division. The authors in [13] compare the performance of six algorithms for cloud coverage taken with sky imagers; the five algorithms are based on thresholding criteria and the sixth utilizes neural network-based cloud segmentation to process the images; their conclusion is that the Hybrid Thresholding Algorithm (HYTA+) and the neural based are producing the best results.

3 Approach

This section presents the design of the techniques for the image segmentation cloud coverage. The proposed Near Infinite Conditions Image Segmentation Algorithm (NICISA) design specifies as input: a) the digital image taken by the sky imager, b) a filtering frame (mask) that is utilized in the process of eliminating the objects at the image margin that are irrelevant to the algorithm such as the buildings, antennas, mountains, etc. and c) the sun location in the image, which is computed based the latitude, longitude, altitude, time, angular offset, camera's focal length, distortion from the camera effects. The following subsections describe first the computation of the sun location in the image, next the filtering process for the unrelated ground objects in the image margin and finally, the image pixels classification as cloudy or cloudless. We note here that the resolution of the low-cost Vivotek camera is 2048x2048 pixels while that of the ASI-16/52 is 1536x1536 pixels and both cameras use 24 bits/pixel (8 bits/channel). Also, we note that the commercial software of the ASI-16/52 system performs defisheye process as the first stage of the cloud cover process, which cannot be deactivated, while all the processes of the NICISA are optimized for the image (both Vivotek and ASI-16/52 provide these files) directly taken from the fisheye lens (without involving any defisheye process).

Using either camera, the development flow for the NICISA software is as follows. The input is the camera image, the time and the geographical location of the camera (latitude, longitude and altitude). Then to accomplish the sun location process, the technique uses the pvlib-python. All these are loaded to the processing unit (either CPU or GPU) by using Python and the PyTorch package to result in the segmented image. The commercially available software that can be used in a complete system with the ASI-16/52 camera is based on the algorithm Schreder and can be executed only in a WindowsR operating system and only on a CPU (not on any GPU).

3.1. Sun Localization

The sun localization problem can be partitioned into two geometric problems with known solutions; a) locating the sun in the sky dome based on the time and location of the observer on Earth and b) projecting a point of a dome in a flat surface, which in this work it is the image frame. The proposed system solves

the first part by using astronomy equations that match time and location coordinates with solar azimuth and elevation. Those equations are implemented in the Pvlib python package [14], a package dedicated to solar power performance simulations. The sun's coordinates on the sky dome can be projected to a flat surface using fisheye equations. The second part of the sun localization problem depends on the projection characteristics of the camera and thus, an ideal fisheye projection is not possible; calibration may be needed in order to find accurate intrinsic coefficients. The number of parameters in this case are 11, which include the orientation of the camera, the sensor's physical dimensions and the 3x3 coefficient matrix of the focal length of the camera. Our solution defines a set of well-known fisheye mapping functions listed in [15]. By defining these mapping functions, we achieve the reduction of the number of hyperparameters to only three: a) the projection function, b) the orientation of the camera and c) a parameter dependent on the focal length of the camera, that is the distance between the lens' centre and the photosensitive chip that captures the image. Finally, the output is a single pixel in the input frame that represents the sun [7].

3.2. Sun Localization

For this process, the proposed solution initially creates a filtering frame of size equal to the input image frame. The PV park engineers decide which pixels will be filtered. Upon the camera installation at the PV park, the engineers check in the input frames the pixels that they will permanently contain ground objects such as trees, antennas, mountains, etc. They check these pixels manually and they create the filtering frame by using the vector graphics software *Affinity Designer* [16]. In this work, we choose to mark the pixels to be filtered as "black" and the remaining in the image as "white."

The processing of the image with the support of the filtering frame first considers whether the image pixel maps to a black or white pixel of the filtering frame. In case that the image pixel corresponds to a black pixel, then this pixel will be ignored for the remaining steps of the image segmentation. If this image pixel corresponds to a white pixel of the filtering frame, then it keeps its initial color and takes part in the segmentation process.

3.3. Image Segmentation Algorithm

For the cloud coverage process we developed the Near Infinite Conditions Image Segmentation Algorithm (NICISA), which is based on the algorithm presented in [8]. The algorithm utilizes for its calculations a very large number (therefore we call it "near infinite"), yet simple conditions. The NICISA is proposed because it is effective with the images taken with either high or low-cost cameras. Moreover, its computational complexity requirements suffice for execution on edge computing devices. The algorithm's output is the classification of the image's pixels either as cloudy or cloudless. The algorithm operates in the following two phases:

1. ***First Phase:*** it divides the image's pixels into these five color sets: green, red, yellow, white and blue. It uses RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value) color spaces for the classification of the pixels into the five aforementioned sets. It also utilizes the distance of each pixel to the pixel that belongs to the sun and the sun altitude. The following steps highlight the sequence of the computations that distinguish the pixels into the five sets:
    a. First, it identifies the green pixels. These are the very bright pixels located close to the sun.
    b. Then it computes the red pixels set, which is a subset of the green set, but all of its pixels are

(rather) cloudless.

c. It determines the yellow set: these pixels are less bright than the green and still brighter than the remaining in the image. These pixels are also located close to the sun (they are part of the solar halo). They may represent either sky or parts of clouds that are bright.

d. It classifies as white pixels those that represent definitely cloudy pixels.

e. It also classifies as blue pixels those that represent definitely cloudless pixels.

2. ***Second Phase:*** it realizes the final classification based again on the solar altitude, the distance between the pixel of the sun and the number of all the other pixels in each of the above sets.

   a. For all those pixels, which were not classified during the first phase in any of the five sets, it will decide which out of the two sets (blue or white) they belong to.

   b. All the green, red and yellow pixels are processed and they are classified as either blue or white.

Figure 1 and Figure 2 show the execution of each of the above steps of the proposed segmentation in the two images taken by the ASI-16/52 and the Vivotek cameras located next to each other. The two images were taken at exactly the same time. Note that in advance of the segmentation process, we have applied the sun localization and the removal of the ground objects processes in both images. For both Figures: in each Figure the first two images are the RGB and HSV color spaces and the following seven images present the results of each of the described above NICISA seven steps. A notable detail in the results is the small crack that appears in all the images (produced by both cameras), at the top of the horizon and which was produced by the process removing ground objects (here it was an antenna) from the input frames.

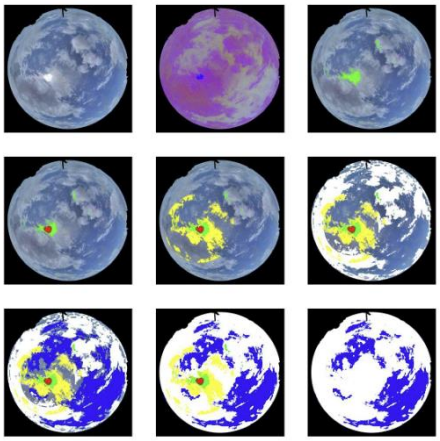


Figure 1: Image Segmentation on the ASI-16/52 [3] input frame.

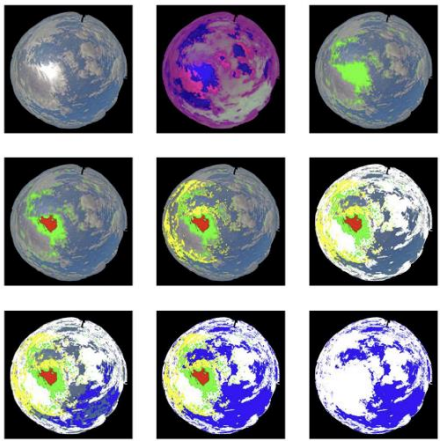


Figure 2: Image Segmentation on the Vivotek [4] input frame.

4 Performance Results

In order to evaluate the performance of the three configurations we tested a variety of weather conditions and cloud coverage. Given the fact that there were not available any datasets evaluated by meteorologists that provide the ground truth, the research team of "Archon" compared the results of the proposed NICISA with the well-established commercial software [3] that supports the ASI-16/52 camera.

Figure 3 presents the results in three indicative cloud coverage cases. We opted for these cases: the clear

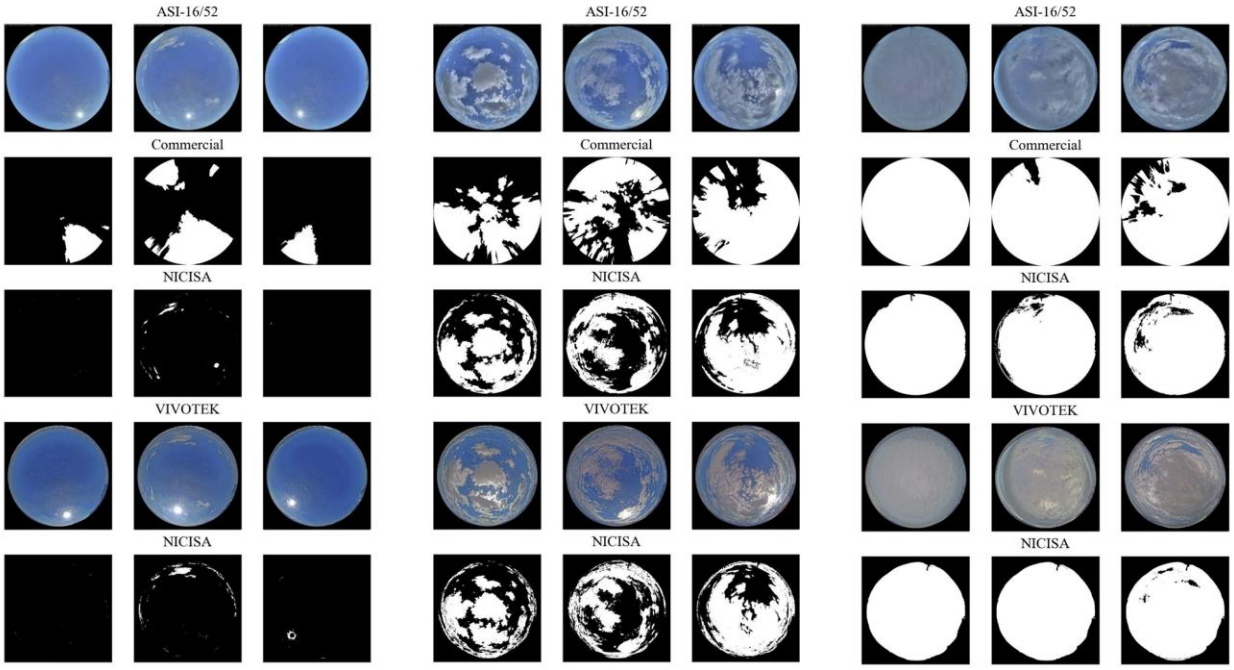
sky (**Figure 3a**), the partly cloudy sky (**Figure 3b**) and overcast (**Figure 3c**). For each of these three cases there are depicted 3 images at different times taken by both cameras. The first line in all subfigures shows the image frame taken with the ASI-16/52 camera. The second line shows the result of the image segmentation of the ASI-16/52 input performed with its commercial software. The third line shows the result of the image segmentation of the ASI-16/52 input performed with the Archon developed NICISA software. The fourth line shows the image frame taken with the Vivotek camera. The fifth line shows the result of the image segmentation of the Vivotek input performed with its NICISA software.

With respect to the real time performance, we note here that according to the controller specifications the PV park cameras provide an input frame every 60 seconds. The following table gives the execution time on the NVIDIA Jetson Nano [2], a widely known edge computing device. The performance is within the time limits of the PV park controller requirements. We also note that the device cannot execute the commercial software.

Table 1. Comparison of execution times between different algorithms.

Algorithm	Camera	Resolution (pixels)	NVIDIA Jetson Nano [2]
Commercial [3]	ASI 16/52 [3]	1536 ×1536	–*
NICISA	ASI 16/52 [3]	1536 ×1536	2.45 s
NICISA	Vivotek [4]	2048 ×2048	4.09 s

***cannot be executed on NVIDIA Jetson Nano**



a. Clear Sky **b. Partially Cloudy Sky** **c. Overcast Sky**

Figure 3: Comparison of Image Segmentation Algorithms for Cloud Cover. (From top to bottom) Row 1: Original image from ASI-16/52 sky imager. Row 2: The result of the commercial algorithm on the undistorted image from ASI-16/52. Row 3: The result of the novel NICISA on the image from ASI-16/52. Row 4: Original image from Vivotek camera. Row 5: The result of the commercial algorithm on the image from Vivotek. White pixels depict cloudy pixels. Foreign objects are masked out.

5 Conclusions and Future Work

This work focused on the comparison of image segmentation solutions for the cloud coverage problem and their utilization in PV parks. The comparison showed that the proposed NICISA set of techniques prevails with respect to the results accuracy and the capability of the execution on an edge computing device. Moreover, it achieves real-time performance. The comparison also showed that low-cost cameras can also substitute sky imagers of considerable cost in geographical locations with mild weather conditions.

Our future work will focus on further improvement of the techniques that the proposed NICISA uses. A possible approach to follow for a fully automated "removing ground objects" will be the correlation among several input frames and detection of objects that remain still over a long period of time. Another significant problem is the cloud coverage prediction and sun irradiance prediction for the next 10-minute window. These processes are of key importance to the PV park and grid management.

References

[1] "Archon," Inaccess Networks, 2022. [Online]. Available: http://archonproject.eu/.

[2] "Jetson Nano," NVIDIA, [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano. [Accessed 26 06 2023].

[3] "ASI-16/52," [Online]. Available: https://www.eko-instruments.com/media/s01jrcy0/asi-16-52-cms-schreder eko-brochure.pdf.

[4] "Vivotek," [Online]. Available: https://download.vivotek.com/downloadfile/downloads/datasheets/fe9382-ehvdatasheet_en.pdf.

[5] S. Park, Y. Kim, N. Ferrier, S. Collis, R. Sankaran and P. Beckman, "Prediction of Solar Irradiance and Photovoltaic Solar Energy Product Based on Cloud Coverage Estimation Using Machine Learning Methods," Atmosphere, vol. 12, no. 3, p. 395, 2021

[6] P. Quentin and L. Joan, A Temporally Consistent Image-based Sun Tracking Algorithm for Solar Energy Forecasting Applications, arXiv, 2022.

[7] E. Papatheofanous, V. Kalekis, G. Venitourakis, F. Tziolos and D. Reisis, "Deep Learning-Based Image Regression for Short-Term Solar Irradiance Forecasting on the Edge," Electronics, vol. 11, no. 3794, 2022.

[8] J. Alonso-Montesinos, "Real-Time Automatic Cloud Detection Using a Low-Cost Sky Camera," MDPI - Remote Sensing, vol. 12, no. Special Issue Assessment of Renewable Energy Resources with Remote Sensing, 2020.

[9] L. Athanasopoulou, A. Papacharalampopoulos and P. Stavropoulos, "Context awareness system in the use phase of a smart mobility platform: A vision system for a light-weight approach," Procedia CIRP, vol. 88, 2020.

[10] D. K. Danso, S. Anquetin, A. Diedhiou and R. Adamou, "Cloudiness information services for solar energy management in west africa," Atmosphere, MDPI, 2020.

[11] A. Kazantzidis, P. Tzoumanikas, A. Bais, S. Fotopoulos and G. Economou, "Cloud detection and

classification with the use of whole-sky ground-based images," Atmospheric Research, vol. 113, pp. 80-88, 2012.

[12] T. Rogiros and C. Alexandros, "Monitoring Cloud Motion in Cyprus for Solar Irradiance Prediction," in Conference Papers in Medicine, 2013.

[13] M. Hasenbalg, P. Kuhn, S. Wilbert, B. Nouri and A. Kazantzidis, "Benchmarking of six cloud segmentation algorithms for ground-based all-sky imagers," Solar Energy, vol. 201, pp. 596-614, 2020.

[14] W. F. Holmgren, C. W. Hansen and M. A. Mikofski, "pvlib python: a python package for modeling solar energy systems," Journal of Open Source Software, vol. 3, no. 29, p. 884, 2018.

[15] B. Felix, "Imaging: Fisheye lenses," WGN, Journal of the International Meteor Organization, vol. 33, no. 1, pp. 9-14, 2005.

[16] "Affinity Designer," Serif, 2022. [Online]. Available: https://affinity.serif.com/en-us/designer/.