

Novel Smartphone Based Energy Calculation of a PV Array Using Hemispherical Images, a Python Script and LTSpice Circuit Simulation.

Björn Fridqvist Nimvik

05/05/2022

This thesis is part of a project which has the aim to create a novel system for energy calculation of photovoltaic arrays that takes both sun irradiation and local shade condition into account. In this thesis a Python script was created and used together with a LTSpice circuit model of a 3×3 photovoltaic array and hemispherical images of the sky to calculate the energy output of the array, when using different amounts of images as input.

Introduction

Shading on a photovoltaic (PV from now on) array can drastically decrease its power output and in turn lower the profits of an installation for the owner. As such conventional methods of only analysing irradiation (while omitting local shade conditions) will overestimate the performance of the array and the profitability of it. Another problem is that most PV installations today are sold with a guarantee of its top effect, rather than an estimation of its full year energy production. This makes it hard for costumers to estimate how much energy they will create over a year and if the investment will be worth the cost. A new method for calculating the energy output of a PV array on site is needed. The system should only require a smartphone, a cheap clip on fish eye lens and the software this project aims to create. A system like this will be beneficial from to all from small privately owned arrays to large scale industrial installations since the profitability could be determined before installment.

The first part of the project was created by a former student at LTH, where he developed a method for calculating solar irradiation, with local shade conditions accounted for, on a very small area using just one hemispherical image captured on a smartphone with a fish eye lens attached [1]. However if we have a large array, can one image represent the whole array? Should we instead take one image in every corner, or do we need to take one image for every cell on the array to accurately represent the actual irradiation over the array?

The goal for this thesis was to create a python script that could take hemispherical images as input, and calculate energy output of a photovoltaic array as output.

Another goal was to analyse how increasing the amount of images affect the output calculation.

Method

The method consists of three parts, designing and building a circuit simulation model of a 3×3 (9 modules) array. Designing and building a Python script that takes hemispherical images of the sky as input, and calculates energy for a PV array as output. When the circuit model and python script were done field tests were performed, using different amounts of images as input for the script.

Circuit model

To perform energy calculations of the array a circuit model in a circuit simulation program is needed. The circuit simulation program used is LTSpice and the array we want to analyse consists of nine modules. The modules placed in a 3×3 pattern, and every module has 60 cells. The cell model used is called a one mechanism 5 parameters model (*1M5P*), and in order to run simulations all 5 parameters must be known. The 5 parameters represents internal attributions of a real cell such as internal resistance in the silicon or the amount of current the cell can create. In total there are 540 cells in the array and all cells must be able to receive their own unique irradiation data, which will be assigned to them by the Python script.

Python script

In the simplest of terms, the Python script takes one or several hemispherical images of the sky as input, and calculates energy for the array as output. See figure (1). The script takes one or many images as input. This step is completely up to the user, but more images means more precise calculations. The images are organized so they are in the correct position relating to the area where a real array would be. Each of the images are then run through shade analysis. The shade analysis creates a shade mask by analyzing the sky view of the image and returns a list of the solar irradiation of every hour of the year, for each image. The irradiation includes direct, diffuse and reflected radiation. The irradiation data is organized in the same way as the hemispherical images, and is assigned to its respective cell in the array. The script then rewrites the information in a way that can be processed by the circuit model. The circuit simulation is prepared, run and the energy output of the array is collected. This is done for every hour of the analyzed time period, and saved into a file on the computer.

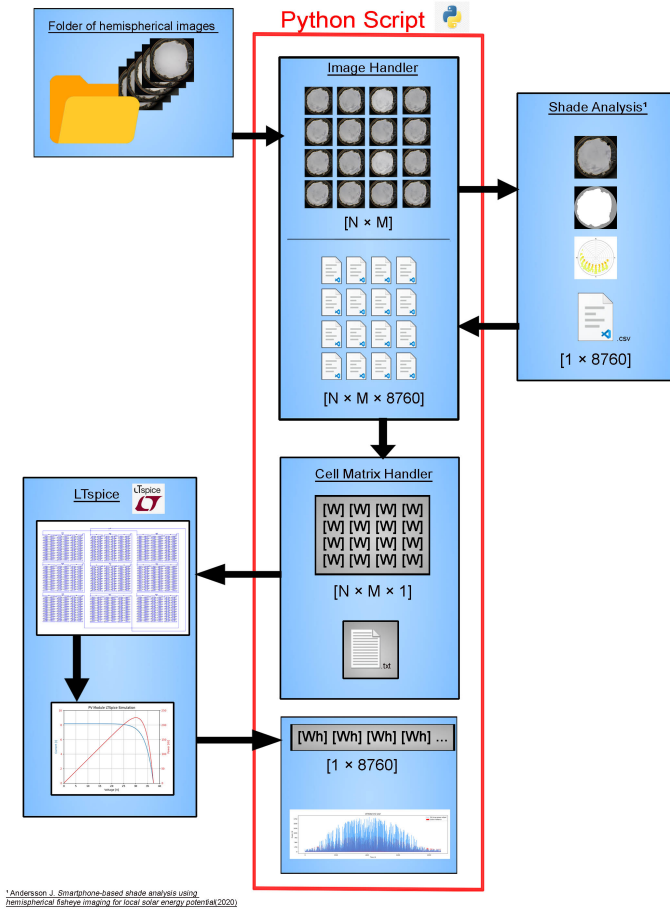


Figure 1: Flowchart of the process.

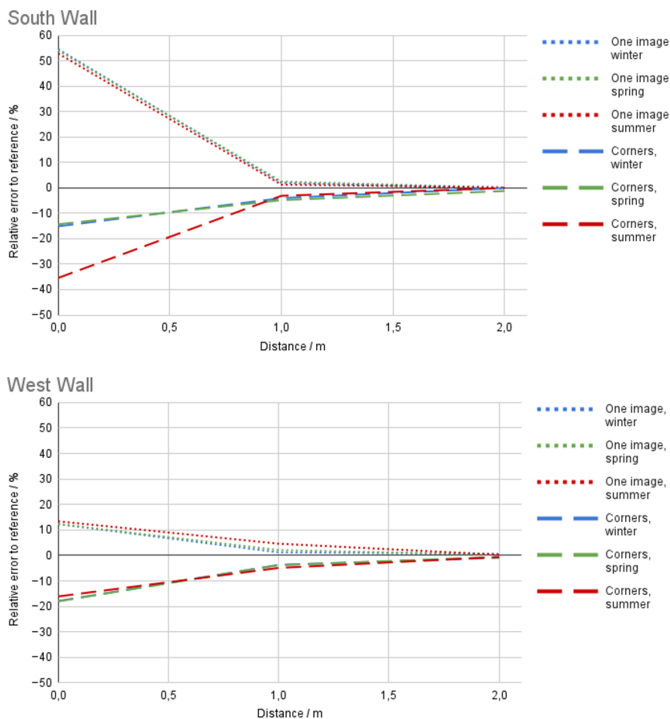


Figure 2: Results from computer generated environments where a one meter high wall was introduced.

Field tests

The purpose of the field tests are to determine if the method in fact works as intended and also to analyze the effect of using different amounts of images as input. Three field tests were performed. One in Lund, Sweden and two in a computer generated environment. The computer generated environment were introduced to test the method under more precise shading conditions.

The first location was at Ulrikedal (55.69, 13.20), Lund, Sweden. A grid was laid out in an area on the ground in the middle of a apartment complex representing the physical dimensions of the 3×3 array. This was done so the hemispherical images could be captured at the correct position on the array. Images were then captured on a smartphone with a fish eye lens attached. 135 images were captured evenly spaced over the grid, with 0° degree tilt and pointing north. The images were later used as input in the script. The field tests were also performed in two computer generated environments to create more precise local shading conditions. Here a one meter high wall was introduced in one case south of the array and in the other to the west. The arrays were placed three at distances from either wall at 0m, 1m and 2m away. Hemispherical images were captured using a tool provided by the 3d modeling program and later used as input in the script.

The question we wanted to answer was how different amounts of images used affect the energy calculation. So instead of only using all 135 images, the script will also be run using two alternative image selection cases. The first alternative is to only use the four images captured in the corners of the array and the second is to only use the one middlemost image of the array.

Result

First we have the results from the Ulrikedal field test. In these field test there was no difference in the energy calculations when using 1, 4 or 135 images as input in the script. 1 image gave the same results as 135 images. We could however confirm that the method works as intended.

The results from the two computer generated environments can be seen in figure (2). The error is defined as the difference in energy calculation when comparing to the case where all 135 images were used as input. The results from the South Wall test gave less error when using four images instead of one when the array was placed at 0m away from the one meter high wall. We also learn that already at one meter away the error in both image cases is less than 5% and at 2 meters away the error is less than 1%.

The results from the West Wall test gave similar results whether using 4 images or 1 image at all distances. At 0 meter away from the one meter high wall the error is about 15% for both image cases, at 1 meter the error is about 5%, and at 2 meter the error is less than 1%.

Conclusion

So did we answer the scientific questions? **Is it possible to calculate the energy output of a PV array using only images and a custom program?** - Yes, it was possible to feed multiple images into a python script and calculate the energy output of a PV array using a circuit model. **How does increasing the amount of images affect the output calculation?"** - While more research is required to fully answer this question we have shown that when the array is placed directly next to a shading object only using one or four images gives a large error. In The South Wall case, using four images gave a better approximation. We have also shown that when the array is placed at 1m away from a 1m high wall the error is less than 5% for any image selection case. At 2m away the error is less than 1%.

Further research

As the goal of this project is to create a consumer friendly product that is cheap and easy to use, the amount of images necessary for a good approximation must be specified at close proximity of an obstacle. For example perhaps there is a better image selection configuration? Maybe a combination of the two image selection cases would reduce the error to a satisfactory point. One image in the middle and four in the corners, 5 images in total. Or perhaps one image per module is better. The LTSpice simulation model should also be simplified in order to simulate larger arrays, as the current model is pushing the limits of the program.

References

- [1] Andersson, J. (2020). *Smartphone-based shade analysis using hemispherical fisheye imaging for local solar energy potential*, Lund University.