Dual-axis solar tracking system based on Raspberry Pi imaging

A Suryanto*, N Hudallah, T Andrasto, C F Adhiningtyas, S A Khusniasari

Electrical Engineering Department, Faculty of Engineering, Universitas Negeri Semarang, Kampus Sekaran Gunungpati, 50229 Semarang, Indonesia,

agusku2@mail.unnes.ac.id

Abstract. Photovoltaic can transform daylight into electrical power. The application of solar tracking system improves the efficiency of photovoltaic system by increasing solar radiation fall on PV’s surface. This paper provides a low-cost solar tracking system based on image processing. Using Raspberry Pi Camera Board as a sensor to eliminate the need of another sensor. Image captured by camera fed to Raspberry Pi to calculate the position of the brightest sky body from the system’s perspective. the proposed system will face to the east and go to sleep mode after the sun set and start to work after the sun rise. This dual-axis solar tracking system based on Raspberry Pi imaging works properly whether it’s a sunny or heavily cloudy day.

1. Introduction

Photovoltaic’s performance can be increased by increasing solar radiation falling on panel, using PV panel cooling and using a smart electronic circuit. One method to increase solar radiation is by using sun tracking system, which in theory can improve solar radiation up to 41% in comparison with static system [1].

There are already researches on development of solar tracking system, both on open loop system without feedback from the sun for example, using an orientation efficiency chart [2], or using sunset and sunrise position to calculate sun path [3], and on closed loop system which receive feedback to maintain accuracy, for example using comparator and LDRs [4][5], using microcontroller and LDRs [6], using Photo Diodes [7][8][9][10], using PC & Image Processing [11][12][13][14][15].

Some research using combined sensors and systems, for example in [15] using both camera and LDRs, in [16] using pyranometer & 2 solar panel as its sensor, in [17] using sun path algorithm and LDRs to correct the path, in [18] using LDR and Gyroscope to track optimal orientation.

By evaluating several solar tracking methods mentioned before, this paper proposes a tracking system which camera applied on Dual-Axis Solar tracker to eliminate the use of another tracking sensor, a Raspberry Pi used to replace PC function in [11], [12], [13], [14], [15]. And by using Raspberry Pi, eliminate the need of external motor logic controller as unlike PCs, it has 40 GPIOs that can directly give logic control to motor driver. Location information needed to calculate sunset and sunrise information like on [3] but used as a sleep time and start time parameter. According to [12] image based solar tracking works perfectly on sunny day and trouble come when the sky is cloudy the accuracy became very low, because incorrect thresholding value and the cloud behave like a noise. In
this paper a simple automatic gaussian filter deviation value and a simple automatic thresholding value applied to correct this error.

2. Design and Method

2.1. Mechanical System
A simple mechanical system can provide smooth enough motion and enough power to move solar panel when using 2 small 77rpm 12v 12watt DC motor. For azimuth orientation, a gearbox 1:6 is coupled to a 77rpm 12v DC motor and for zenith orientation, a 10mm screw rod coupled to a 77rpm 12v DC motor. This mechanical system provides a slow, smooth and powerful enough movement for 2x20Wp solar panel. Material for frame construction is 30mm*30mm square hollow section steels, and 30mm iron flat bars. The construction model & prototype shown on figure 1.

![Figure 1. Solar Tracker Construction Model (Left) and the Built Prototype (Right)](image)

For Raspberry Pi Camera Board to be able to handle outdoor use and intense sunlight, a special case and filtering are needed. The case made from Pertinax board, since it’s lightweight, durable and high temperature resistance, the top of the case made from shade 10 DIN welding glass to protect camera from intense light and UV. A wide-angle lens applied to Raspberry Pi Camera to increase its field of view. Figure 2. show simple figure for the camera casing.

![Figure 2. Camera Casing Model](image)
2.2. Electrical System
This solar tracker is designed with self-sustain capability in which external power doesn’t need supply as long as it gets enough solar energy and its part on a good condition. To be able doing this the electrical system separated into 4 groups: Power Sector, Sensor Input, Control Unit and Actuating System. The electrical system’s block diagram shown on figure 3.

![Diagram Block of the Electrical System](image.png)

Figure 3. Diagram Block of the Electrical System

2.2.1. Power Sector.
Power sector’s function is to store energy collected by solar panel and regulate it so it can be used for powering control unit and actuating system. This power sector consists of solar charger, battery and power regulator. For 2x40Wp solar panel, it needs minimum 10A solar charger, a 10Ah 12V lead acid battery is enough to store energy when the sun present and supply energy while the sun is set for the whole system without external device attached, to regulate 5,1V for Raspberry Pi and the sensors a 5V 5A DC-DC Buck Converter is needed and modified by attaching fuses and battery discharge/undervoltage safety circuit, and adjusting voltage output to 5,1V by adjusting the trimmer resistor, while the 12v appliances only take the output of the discharge/undervoltage safety circuit assuming the battery voltage won’t be higher than 14V.

2.2.2. Sensor Input.
Sensor input consist 4 Limit switches for position limit, RTC module for time input, Function Button act as user input and Pi Camera for the main sensor. The 4 limit switches will limit the movement of the panel so it won’t overpull and strangle the wires. Function button function as input to start the system for the first time and to stop the system in case of maintenance needed. Pi Camera Board is the main sensor who feed sky image to Raspberry Pi.
2.2.3. Control Unit.
The main control unit is Raspberry Pi assisted by a limiter circuit since sometimes Raspberry Pi is not fast enough to process limit switches and stop motor in time while 1 second control lag is enough to cause problem to actuating system. The limiter circuit is an array of DPDT relays which cut the connection of one of the Motor Driver’s input from Raspberry Pi’s output when the corresponding limit switch is active.

2.2.4. Actuating System.
The actuating system consist 2 DC Motors, a Double H-Bridge Circuit for motor driving, a 12v Fan and its driver circuit.

2.3. Tracking Method
For sun tracking purpose, in general there are 6 steps: OpenCV & camera initialization, transform captured RGB data to a 2-dimensional Grayscale data, smoothing and filtering from noises, calculate sun’s parameter, calculate automatic gaussian filter and automatic thresholding value for next capture, begin turning PV. These steps written on Python language to make it easier to program the Raspberry Pi.

2.3.1. Step 1. OpenCV and camera initialisation started by importing the library of PiCamera, PiRGBArray & CV2, followed by declaring PiCamera() data in to a variable and set iso & exposure to automatic, set frame rate to 15 fps & scale down resolution to 320px*240px to reduce Raspberry Pi’s workload. Make a random odd number between 1-9 as initial value variable for sun’s radius, for example fill 7 to variable R3. Rotate the image data if necessary, to make it easier to program, make sure the horizontal data location (x coordinate) is equal to azimuth orientation and vertical data location (y coordinate) is equal to zenith elevation.

2.3.2. Step 2. To transform RGB data captured from camera to Grayscale in Python, use cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) line. By inserting its data to a variable, for example gray image, a 2-Dimension Grayscale image will be created, which each pixel represents a brightness value.

2.3.3. Step 3. Apply Gaussian filter to filter out the grayscale image to eliminate noise and smoothing sunlight’s edge & spikes so it become more circular. The first frame captured have to use the predetermined variable R3 (sun’s radius variable) as gaussian filter x & y deviation value, while deviation value for next frame will be calculated on step 5.

2.3.4. Step 4. Calculate the sun’s parameter by finding position and value of the pixel with max value. This information can be found by using cv2.minMaxLoc() function which return the value and position of a single pixel with highest value and a single pixel with lowest value. Then put value and position of the pixel with highest value to each variable, for example max val for value, xmax for horizontal location, ymax for vertical location.

2.3.5. Step 5. Then use 75% of the value of the pixel with the highest value as thresholding value. By using this, thresholding value will automatically adapt to sky condition no matter how fast the weather change. Then calculate the approximation of sun’s radius from the current threshold area, use it as next frame’s gaussian filter deviation value by filling the radius approximation to R3. But be careful, because deviation value is odd number only, even number will bring an error and stop the program, therefore if the radius is even number it must be reduced by 1. By assuming the sun is always close to circle, the radius calculation equation will be expressed as:

\[
R = \sqrt{\frac{A}{\pi}} \quad f(R) = \begin{cases} 
R \% 2 = 0, & R3 = R - 1 \\
R \% 2 = 1, & R3 = R 
\end{cases}
\]  

(1)
Where \( R \) is calculated radius, \( A \) is area from threshold image, and \( R3 \) is radius used for Gaussian filter deviation value. The calculated thresholding & gaussian filter deviation value will be used for next frame’s processing value and will be recalculated after 1 second (around 15 frames) so the recalculation process won’t overburden the processor.

2.3.6. Step 6. For panel turning, we first need to make a point in the middle of image’s resolution and put it in a variable to point out this centre point’s horizontal and vertical, for example \( x_{centre} \) and \( y_{centre} \). Then this process depends on \( x_{max} \) (horizontal position of brightest pixel) and \( y_{max} \) (vertical position of brightest pixel) calculated from step 4. Compare the value of \( y_{centre} \) and \( y_{max} \), if the value of \( y_{max} \) is more than 5 points under \( y_{centre} \) turn panel downward until \( y_{max} \) is between +/- 5 point of \( y_{centre} \), if the value of \( y_{max} \) is more than 5 points above \( y_{centre} \) turn panel upward until \( y_{max} \) is between +/- 5 point of \( y_{centre} \).

The same is applied to \( x_{centre} \) and \( x_{max} \), but with addition of automatic motor rotation correction. This automatic correction works by reversing motor polarity if the \( x_{max} \) moves away from \( x_{centre} \) instead of move closer when try to rotate left or right. This move away rotation happened after the panel vertically flipped past noon, making left-right movement needs to be swapped. By doing this automatic rotation correction, the cabling can be swappable because it will be corrected by the program.

2.4. Time-Based Behaviour
While the sun set, the system needs to rotate back facing east to prepare itself for the next day. After the sun sets, the system will have to lower the power consumption so the power stored on battery can last until the sun rises. When the time for sun rise come, the system will start to works normally. To do this, an exact time of sunset and sunrise needed and this time is dependent on the location since different location have different sunset and sunrise time. This local solar time can be calculated using Python library named Astral. By providing location data, latitude, longitude, elevation and time zone, Astral will calculate solar time of provided location point. Then the system will compare current time with calculated time, when current time is the same or later than sunset time, the system will immediately rotate back facing to east. After that the program will enter a loop program without taking frames from camera and not activating any output. When current time is the same as sunrise time, first system will check if panel orientation is facing east or not, if not the system will rotate to east, after its confirmed that panel is facing east the system will break form the loop and start to works normally. This time-based behavior has a higher priority than tracking process and will interrupt and stop the tracking process when the sun sets.

3. Result
This solar tracker system consists of power sector, sensor input unit, control unit, and actuating system. Where power sector consists of solar charger, battery, and power regulator. Sensor input unit consists of 4 limit switches, raspberry pi camera board, function button, and RTC Module. Control unit is a Raspberry Pi 3 model B assisted by a limiter circuit board to ensure the actuation output start and stop in time in case the Raspberry Pi not stop it in time. the output actuation system consists of a 12v motor connected to a 10mm screw rod for zenith rotation, a 12v motor connected to 6:1 Gearbox for azimuth rotation, and a 12v DC fan to prevent system overheat.

The experiment takes place at Semarang City on 28 – 30 September 2019. The weather is mostly sunny with sometimes heavy cloud covering the sky. Even with the sudden change in the sky body this solar tracker system can works excellently. The figure 4. shows the captured image where a is 4a is when the sky is clear, 4b when the sky is cloudy. The tracked brightest region (the sun) is marked by yellow circle that change it size based on calculated radius R3 and green pixel with 3 pixel radius indicate the center of the brightest region. The text on these images are written automatically by the program for data logging purpose. The top line is solar time (sunrise, noon, sunset), the third line from the bottom are sun coordinate and date, the second line from the bottom is the action being carried out,
the most bottom line is minimum and maximum pixel value, current time, CPU temperature and calculated radius.
4. Conclusion
The energy source of solar panel is light and the most powerful light source is the Sun and no other light source stronger than the sun on a day light. Thus, the most appropriate method for solar tracking is to detect the brightest region in the sky by using a camera. But the camera needs a shield to protect it from the immense of the sun’s light. And previous brightness-based solar tracking having a problem with no practical thresholding value, noise problem by the clouds and gaussian filter deviation value. This tracking system can solve that problem. The camera protected by welding glass, image thresholding value and gaussian filter automatically corrected so the clouds will blend with the background if behave like a noise. This makes the proposed solar tracking system more robust and can be practically applied in bigger scale. However, there are plenty thing needs to be improved like the mechanical system, the processing unit can be replaced with a faster and stronger unit or a more energy efficient unit.

References
[1] Cotfas D T & Cotfas P A 2019 Multiconcept methods to enhance photovoltaic system efficiency International Journal Of Photo Energy 2019
[2] Jose R et al. 2019 PV tracking design methodology based on an orientation efficiency chart Appl. Sci. 2019 894-909
[3] Syaffi, K. M Nor, M Abdel-Akher 2010 Grid-connected Photovoltaic Models for Three-Phase Load Flow Analysis, 3rd IEEE International Power andEnergy Conference PeCon
[4] Iqbal M K, Islam T, Chowdury M, Imteaj A 2015 Construction of single axis automatic solar tracking system International Journal of u- and e- Service, Science and Technology 8 389-400
[5] Wang J-M and Lu C-L 2013 Design and implementation of a sun tracker with a dual-axis single motor for an optical sensor-based photovoltaic system Sensors 2013 3157-68
[6] Abdullah S K 2018 The powerful activity and sensitivity for the sunlight tracker based on the arduino platform Journal of Babylon University/Pure and Appl. Sci. 26 191-200
[7] Merlaud A, De Mazière M, Hermans C, Cornet A 2012 Equations for solar tracking Sensors 2012 4074-4090
[8] Morón C, Ferrándiz D, Saiz P, Vega G and Diaz JP 2017 New prototype of photovoltaic solar tracker based on arduino Energies 2017 1298 – 1311
[9] Tudorache T and Kreindler L 2010 Design of a solar tracker system for pv power plants Acta Polytechnica Hungarica 7 23-39
[10] Khadidjaa B, Dris K, Boubeker A, Noureddine S 2014 Optimisation of a Solar Tracker System for Photovoltaic Power Plants in Saharian region, Example of Ouargla, The International Conference on Technologies and Materials for Renewable Energy, Environment and Sustainability Energy Procedia 50 610-618
[11] Garcia-Gil G and Ramirez JM 2019 Fish-eye camera and image processing for commanding a solar tracker Heliyon 5 01398
[12] Wei C-C, Song Y-C, Chang C-C, and Lin C-B 2016 Design of a solar tracking system using the brightest region in the sky image sensor Sensors 16 1995-2006
[13] Ruelas A, Velázquez N, Villa-Angulo C, Acuña A, Rosales P, and Suastegui J 2017 A solar position sensor based on image vision Sensors 2017 1742 – 1755
[14] El Kadmiri Z, El Kadmiri O, Masmoudi L, and Bargac MN 2015 A novel solar tracker based on omnidirectional computer vision Journal of Solar Energy 2015 149852
[15] Azizi K and Ghaffari 2013 Design and manufacturing of a high-precision sun tracking system based on image processing International Journal of Photoenergy 2013 754549
[16] Grofu F 2018 Control system for photovoltaic panels tracker Fiabilitate si Durabilitate 2018

Figure 4. Solar Tracker Captured Image, A. When the Sky Clear. B. When the Sky Cloudy
333-338

[17] Ferdaus R A, Mohammed MA, Rahman S, Salehin S, Mannan MA 2014 Energy efficient hybrid dual axis solar tracking system *Journal of Renewable Energy* **2014** 629717

[18] Pulungan AB, Son L, Syafii 2018 A Review of Solar Tracking Control Strategies *Proceeding of EECSI* 319-323