INTELLIGENT SYSTEMS AND SUSTAINABLE ENERGY
E-ISSN: 3154-7532
VOLUME 1, ISSUE 1, 2026, pages 19 – 33
DOI: https://doi.org/10.15282/isse.1.1.2026.13883

RESEARCH ARTICLE

Advanced image-processing-based single-axis solar tracking for enhanced photovoltaic performance

Mohd Shawal Jadin[1*], Sabhan Kanata[2], Mohd Herwan Sulaiman[1], Mohd Redzuan Ahmad[1], Kamarulazhar Daud[3]

[1]Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600, Pekan, Pahang, Malaysia
[2]Faculty of Engineering and Computer Science, Universitas Muhammadiyah Semarang, Semarang, Central Java, 50273, Indonesia
[3]Electrical Engineering Studies Universiti Teknologi MARA, Cawangan Pulau Pinang, Permatang Pauh, Pulau Pinang, Malaysia

Abstract - Solar photovoltaic (PV) energy generation is strongly influenced by the angle at which solar irradiance strikes the PV module surface. Conventional fixed-tilt PV systems cannot track the sun's apparent daily movement, resulting in reduced energy yield. Additionally, many sensor-based solar tracking systems experience performance degradation under non-uniform sky conditions. This study details the design, implementation, and experimental evaluation of an image-processing-based single-axis solar tracking system intended to improve PV performance while preserving mechanical simplicity and cost efficiency. The system that controls these panels should be perpendicular to the sun's rays to obtain the maximum electricity using the image processing method. When the image is received, it will be processed through several image processing techniques to produce the sun's exact location. While moving the PV panel, a vertical actuator is used with a stepper DC motor to control it, and a gyroscope sensor is used to detect the PV panel tilt angle. Raspberry Pi is used as the main processor. Several data have been taken to analyse and validate the system. By dividing into sunny day cases and cloudy day cases, the system is tested by two methods, which are the brightest region method and the contour finding method. The result shows that each method performs well on sunny day cases, with more than 71% of data detecting the actual angle with an error below 15%. The result for cloudy day cases shows both methods have a higher error number of data, which is only 27% error, below 15% for the brightest region, and 63% for contour finding.

Article history
Received : 23 January 2026
Revised : 24 February 2026
Accepted : 2 March 2026
Published : 4 March 2026

Keywords
Solar tracking system
Single-axis photovoltaic tracker
Image processing
Computer vision
Renewable energy systems

1. Introduction

The growth in energy demand, increasing concerns about climate change, and the depletion of fossil fuel resources have caused a global shift toward sustainable and renewable energy technologies [1]. Among renewable energy sources, solar energy is the most promising option due to its abundance, environmental advantages, and the continual decrease in system costs. Consequently, photovoltaic (PV) systems are now widely deployed in residential, commercial, and utility-scale applications. However, during operation, the electrical performance of the PV array is highly dependent on the tilt angle, making the optimal orientation paramount in order to maximize the power as well as the energy production [2]. Traditionally, the fixed-tilt PV installations are typically oriented based on geographic location and season. While this approach looks like a simple system design, it nevertheless fails to accommodate the daily sun's movement, resulting in energy losses in the morning and afternoon [3]. One of the promising solutions is to use solar tracking systems by continuously adjusting PV module orientation to keep them nearly perpendicular to the sun throughout the day. It is shown that numerous studies have demonstrated that the solar tracking systems can increase the energy output by approximately 25–38% compared to fixed PV systems, depending on climate and tracking configuration [4]. Generally, solar tracking systems can be divided into passive or active. In addition, they can also be classified into single-axis and dual-axis. Dual-axis trackers move in both azimuth and elevation, increasing solar capture, but need more mechanical parts and cost more. This makes them less practical for small or medium PV system installation. Single-axis solar tracking systems can improve performance with simpler designs, making them more practical and cost-sensitive [3]. From a recent report, it is found that most active single-axis solar tracking systems use light-based sensors, such as light-dependent resistors (LDRs) or photodiodes, in order to determine the sun's position. Although these sensor-based methods are simple and cost-effective, their accuracy is susceptible to sensor mismatch, intense sunlight, as well as missing the correct position, especially during cloudy conditions [7]. To improve these limitations, time-based and astronomical tracking methods have been proposed, which utilize time/date and geographic location for calculating the sun's position [8]. Although these approaches eliminate the need for light sensors, they lack the real-time feedback and are vulnerable to installation or alignment errors that can lead to tracking inaccuracies [9]. Recent developments in computing and imaging technologies have enabled image-based tracking systems that estimate the sun's position using image processing methods. This method offers better sun position detection under variable sky conditions and can reduce dependence on the use of

***CORRESPONDING AUTHOR | Mohd Shawal Jadin** | ✉ mohdshawal@umpsa.edu.my

light sensors [10]. Nevertheless, many image-based systems continue to rely on dual-axis configurations or require complex hardware, increasing costs and restricting widespread adoption [2].

Passive solar trackers use mechanical or thermal effects, such as fluid expansion or shape-memory materials, to move in response to solar heating. Although passive solar trackers are simple and require no electricity, however it responds to slow and lack precision for the accurate position of the PV array [11]. In contrast, active solar trackers, favored in contemporary PV systems, employ electric actuators managed by sensors, microcontrollers, or algorithms, offering better accuracy and flexibility. The tracking system can be single-axis or dual-axis. Single-axis trackers rotate PV panels along a single axis, typically east–west or north–south, and are popular for their simplicity, lower power consumption, and lower installation and maintenance costs. Research indicates that single-axis tracking can increase PV output by approximately 20–30% compared to fixed systems, depending on geographic location and climate [6]. Early single-axis trackers primarily relied on sensor-based control using LDRs or photodiodes to adjust panel orientation. While easy to implement, these systems are sensitive to environmental changes; sensor mismatches, intense sunlight, or cloudy weather can cause tracking errors and unstable operation [12]. Time-based and astronomical tracking methods were then introduced, utilizing time, date, and location to follow the sun's movement. Although these approaches consume energy and are unaffected by sunlight variability, they lack real-time feedback and may be prone to errors from misalignment [13]. Dual-axis solar trackers can move in both azimuth and elevation, resulting in superior tracking accuracy and potential energy output increases of 30–40% compared to fixed PV systems. Consequently, they are frequently deployed in applications where maximizing energy yield is critical, such as concentrated solar power and large-scale PV plants [11]. However, dual-axis trackers require additional actuators, sensors, and control units, which increases costs, power consumption, and mechanical complexity. These factors contribute to higher maintenance demands and may reduce reliability, particularly in outdoor environments. For small- and medium-sized PV systems, incremental energy gains may not justify the additional costs [14]. Hybrid tracking systems integrate sensor-based and time-based approaches, typically employing time-based control with supplementary light sensors for correction. Although hybrid systems offer greater robustness under variable sky conditions, their increased complexity and energy consumption can diminish their overall benefits [15][16]. Furthermore, hybrid and sensor-assisted trackers remain susceptible to tracking errors caused by uneven sunlight due to clouds, atmospheric effects, or reflections [17].

Advancements in computing, affordable cameras, and computer vision have enabled image-based solar tracking systems. These systems use sky images and simple image-processing techniques—such as brightest-region detection, contour extraction, and centroid calculation—to determine the sun's position [11]. This approach offers improved angular accuracy, less risk of sensor degradation, and better performance in partly cloudy weather. It works best when the sky is clear. However, many image-based trackers use dual-axis setups or require extra sensors, such as GPS or motion sensors, which add complexity and cost [6]. Performance also drops under heavy cloud cover, when clouds and reflections hide the sun. Solving this problem will need better algorithms, improved filtering, and thorough outdoor testing [18]. The existing literature shows that traditional single-axis sensor-based solar trackers perform poorly under changing sky conditions. Dual-axis and hybrid systems work better but are more complex and expensive. While image-based tracking has potential, it is rarely used in single-axis trackers. This gap reveals the need for a cost-effective, accurate, reliable, and simple image-based single-axis tracker. This study addresses the need by adding computer vision to a single-axis system and by testing two image-based sun-detection methods under both sunny and cloudy conditions. The goal is to offer a practical and scalable way to improve PV energy capture. The system uses a wide-angle camera to photograph the sky. Image processing with OpenCV on a Raspberry Pi estimates the sun's position. Two image-based techniques—brightest-region detection and contour-based segmentation—are tested outdoors under different weather conditions. The sun's position is converted into tilt commands for a linear actuator, and a gyroscope provides real-time feedback for closed-loop control. This work makes three main contributions. First, it describes the design and construction of a low-cost, image-based single-axis solar tracker. Second, it experimentally compares two methods for image-based sun detection. Third, it analyzes performance under different sky conditions. The results aim to demonstrate a practical, scalable alternative to traditional sensor-based trackers that supports better PV energy capture in real-world use.

2. Materials and Methods

The system uses a wide-angle camera as its main sensor to capture images of the sun. A Raspberry Pi runs an image-analysis algorithm to find the sun's position in these images. Based on this, the system sends commands to a linear actuator to adjust the tilt of the photovoltaic (PV) module. As shown in Figure 1, the setup includes a Raspberry Pi as the main processor, a wide-angle camera for sky images, and a gyroscope sensor to measure the module's tilt. The DC linear actuator, which is attached to the PV module, gets control signals from the motor driver module based on the Raspberry Pi's output. After each image is captured, the data goes to the processing unit for analysis. At the same time, the gyroscope continues to report the PV module's current tilt. The system compares this real-time tilt angle with the target angle from the image processing. If there is a difference, the actuator adjusts the module's orientation. If the angles match, no adjustment is made. This feedback loop keeps the PV module aligned with the sun's path during operation.

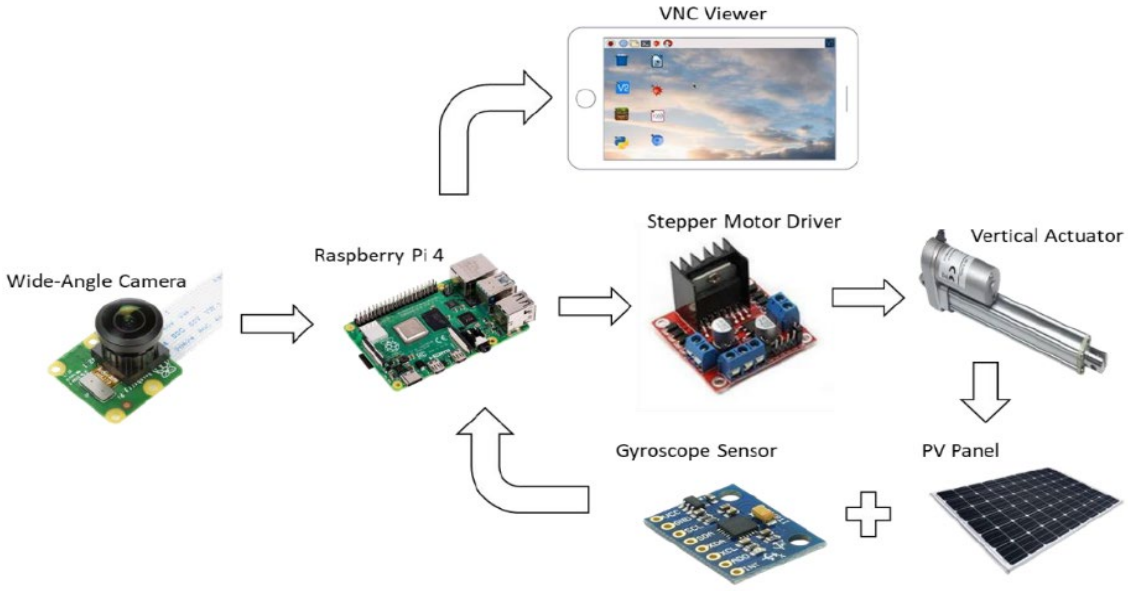
The system uses only two sensors, which are a camera module and a gyroscope-based tilt sensor. A stepper motor controller, powered by a 12 V battery, to run the actuator. All sensors and actuators are connected to the Raspberry Pi, which controls the system. The single-axis tilt sensor (SN-ENC03R) gives real-time tilt measurements with an analog output, making it easy to connect. It has a low-noise amplifier for better sensitivity and reliability, runs on a +5 V supply, and includes a 3.3 V voltage regulator. The sensor has a raw sensitivity of 0.67 mV/°/s, a full-scale range of 300°/s, and an amplifier gain of 4.7, allowing for accurate monitoring of the PV module's orientation. The wide-angle camera module

connects through the CSI interface and uses an improved optical design with a fixed LS-40180 fisheye lens. It captures high-resolution images (2592 × 1944) and records 1080p video at 30 fps. An IR filter is built in to improve image quality in outdoor light. The camera module measures 36.0 mm × 36.0 mm and helps ensure reliable solar tracking.

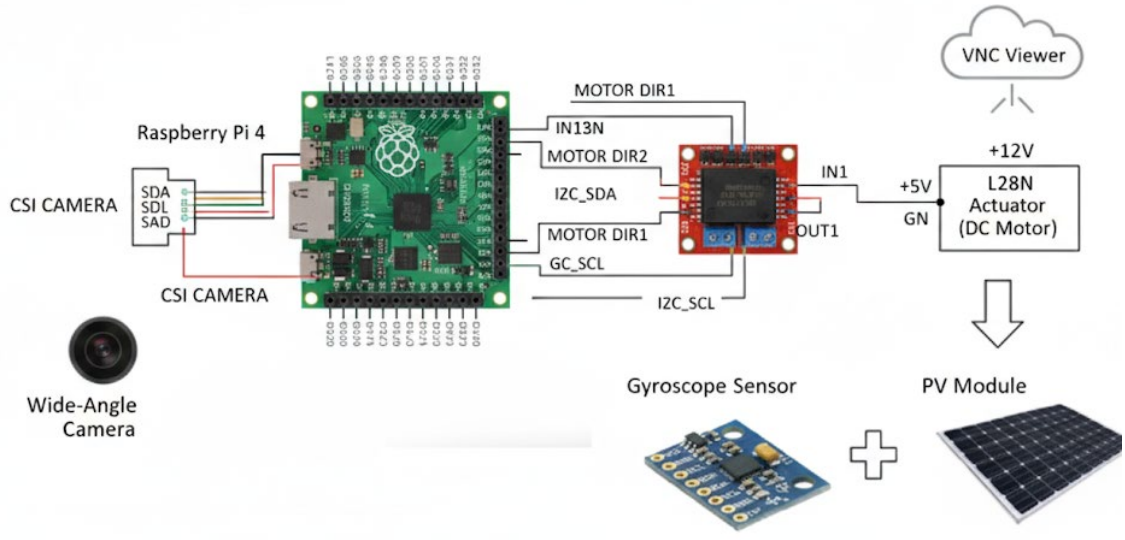
An L298N DC motor driver module controls the linear actuator that moves the PV module's support structure. The motor driver works with input voltages from 3.2 V to 40 V and usually uses a 5 V supply. The linear actuator runs on 12 V DC, has a power rating of 25 W, and draws 3 A of current. It moves at 10 mm/s, can exert up to 700 N of force, and extends up to 500 mm, allowing for precise, stable tilt adjustments. The single-axis PV tracking system follows the workflow shown in Figure 2. First, the camera takes a sky image, which is processed to find the sun's center. The system then calculates the desired tilt angle for the PV module. The gyroscope sensor measures the actual tilt angle, and the system compares it to the target angle. If there is a difference, the actuator adjusts the module. If the angles match, no adjustment is needed. This cycle repeats to keep the PV module aligned with the sun throughout the day.

2.1. Image Segmentation and Sun Position Identification

To identify the sun's position using the image processing method, the process is presented in the flowchart in Figure 3. The first step after capturing the image is to convert it from RGB to greyscale. Then, after it changes into greyscale, it will be filtered using a Gaussian smoothing filter and converted to a binary image using the binary thresholding method. After that, the morphological method makes the image smoother to ensure the next step is successful. The next step is contour finding and automatically computing the contour's centre. The next step is drawing the contour and detecting the centre.



(a)



(b)

Figure 1. A constituent part of a single-axis PV tracking system

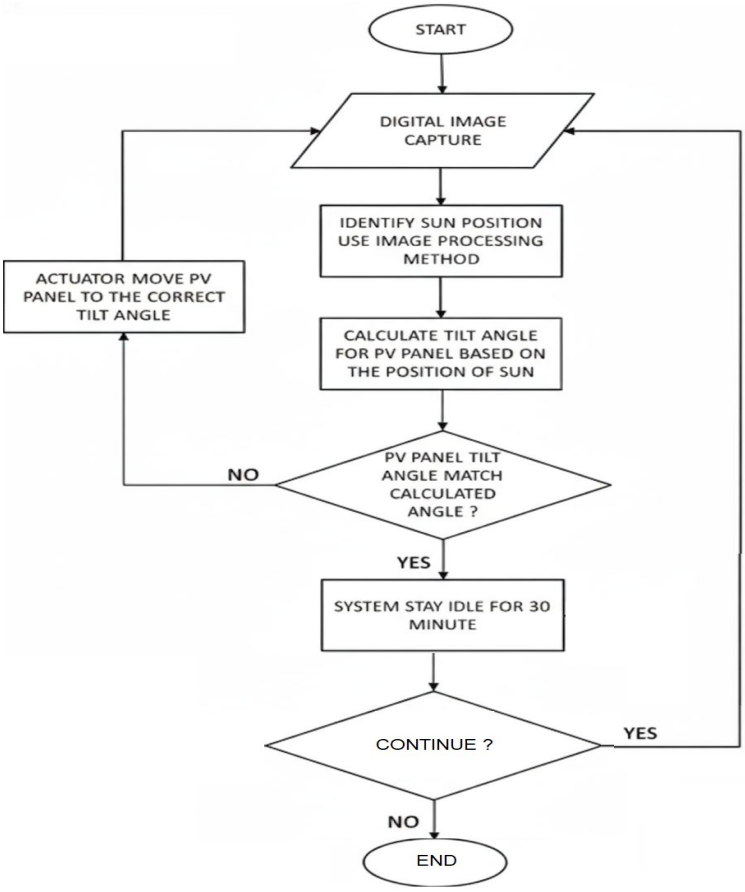


Figure 2. Flow chart of the system

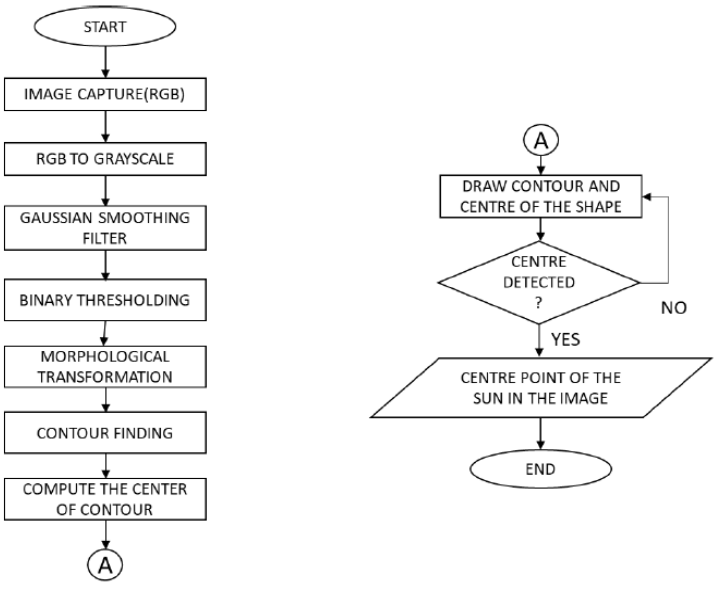


Figure 3. Flowchart of the image processing method

OpenCV is an open-source computer vision library for real-time computer vision applications. In this paper, several techniques are tested to ensure the results are precise, since the objective is to determine the exact sun location. The grayscale denotes the image, transforming it from the RGB to the YCbCr colour space. The brightness of each colour represents a value between 0 and 255. The conversion relationship between the grayscale and RGB is listed in Eq. (1) [17]. Y is the luma component, and Cb and Cr are different red and blue chroma components. YCbCr colour spaces are derived by the mathematical coordinate transformation that is nonlinearly encoded based on gamma-corrected RGB colour spaces. Figure 4(a) shows the colour of the sky image, and Figure 4(b) shows the grayscale image derived from the RGB image.

$$Y = 0.299R + 0.587G + 0.114\ B \quad (1)$$

The picture will have some high-frequency noise for the sun's image, and a low-pass filter is necessary to remove it. In this research, the technique used is the Gaussian smoothing filter as the low-pass filter that uses a Gaussian filtering G(x,y), in the following used to blur the image and remove noise and other extraneous noise. This function in OpenCV is as follows: cv.GaussianBlur(image,(5,5),0). The value of the filter can be adjusted as an odd value only. Figure 5 shows the image after applying this technique.

2.1.1. Thresholding-Based Region Segmentation

In digital images, thresholding is one of the most straightforward techniques for segmentation. The thresholding function can be used from the grayscale image to create a binary image—the function cv. The threshold is used to apply thresholding; the first argument needs a grayscale image source. The second argument is the threshold value used to classify pixel values. The third argument is the maximum value assigned to the pixel that exceeded the thresholding. The chosen method for this work is Binary Thresholding. Below is the thresholded image, as shown in Figure 6. Morphological transformation is an operation based on the image shape. This operation will use two images: the original image and the structuring element, which will determine the operation. This work will use the variant form of this technique: the opening-and-closing function. This technique either creates or removes a small white dot or removes noise from our images, inside and outside the object. This process will give a better image for another operation.

To get a binary mask suitable for contour extraction, a threshold at a fixed value was applied to the grayscale image. Through the method of trial and error to test across all the sunny and cloudy sample images, a threshold value of T = 200 (on a 0–255 grayscale range) was found to provide the most reliable separation between the bright solar region and the surrounding sky background. This value consistently isolated the solar region while minimizing noise from cloud edges and diffuse scattering, and therefore offered the best trade-off between image segmentation accuracy and algorithmic robustness. The chosen threshold was validated through visual inspection of all images and was retained for all experiments to ensure methodological consistency.

To find the centre of the object, the centre of the contour needs to be computed, and after that, the contour and the centre are drawn. This operation will join several operations. The first function is cv.moments(contours[i]) to find the object in images and then compute the centre. After that, the contour will be drawn by the function cv.drawContours(image, [c], -1, (0, 255, 0), 2). Figure 7 is the result of this operation. For tilt angle, β, the error % accepted is set ±15% and below, which is not more than 27° from the expected angle (α). Based on previous studies, a 15° change in tilt angle, either west or east, results in less than a 1% reduction in the total solar radiation received by the PV module. Therefore, pixel acceptance error % will be set to ±10%.

(a) (b)

Figure 4. (a) RGB and (b) grey image of the sky

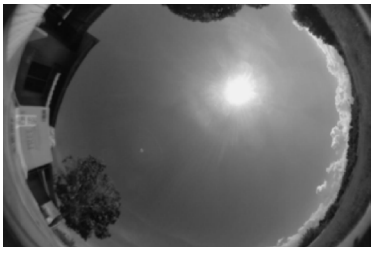
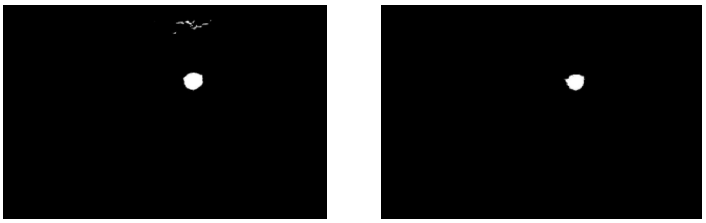


Figure 5. Image after applying Gaussian filter



(a) (b)

Figure 6. Segmented image of the sun and image after morphological transformation

2.1.2. Region Segmentation Based on Highest Pixel Intensity

For this method, the function used is cv.minMaxLoc. This function finds the brightest pixel in the entire image and then circles around that pixel. This function requires the image to be converted to grayscale and cannot be a binary image. The result is shown in Figure 8. This method is sensitive to noise; thus, a filter is needed.

2.2. Tilt Sensor and Actuator

The tilt sensor used in this work is SN-ENC03R. This sensor is attached to the module, which will be the PV module's position. This sensor can track and compare the module's actual angle with the desired tilt angle value. The system will compare the desired tilt angle from the image processed, and the tilt sensor will detect the actual tilt angle to move the motor later. After the system compares the desired tilt angle with the actual tilt angle, the actuator will move at the specified speed to rotate the PV module to face the sun. This actuator has been set at a minimum value of 25° and a maximum of 140°. Once the value is set, the actuator will stop. The following day, when the sun rises, the module's position will move the PV module back to 40° and start the operation over.

2.3. Mapping the Image Centroid to the PV Tilt Angle

Once the solar disc (or its bright region) is segmented and the image centroid $(u_s, v_s)$is obtained (Section 3.1), the tracker must convert this 2-D pixel coordinate into a single-axis tilt command β for the module. This section details (i) the camera model and calibration, (ii) pixel-to-ray conversion, (iii) projection onto the tracker's rotation plane, and (iv) actuator/IMU feedback to realize the commanded tilt. The mapping is implemented on the Raspberry Pi using the wide-angle (fisheye) camera described earlier and executed in closed loop with the gyroscope-based tilt sensor and the DC linear actuator.

2.3.1. Camera Model and Calibration

Because the camera uses a fixed LS-40180 fisheye lens (2592×1944 native resolution), we adopt a fisheye projection rather than a simple pinhole model. Calibration is performed once (or when the camera/optics are remounted) using a checkerboard sequence to estimate the intrinsic parameters K and the fisheye distortion coefficients d (OpenCV fisheye model). The extrinsic transform (R_{cw}, t_{cw})between the camera and the tracker frame is then estimated via a simple two-pose procedure (explained below). This calibration ensures that pixel measurements map to correct sight rays despite the fisheye optics.

- Intrinsic model (equidistant fisheye):
  For a direction unit vector $r = [r_x, r_y, r_z]^\top$in the camera frame, the ideal (undistorted) projection (x_u, y_u)on the normalized image plane is

$$\theta = \arccos\left(\frac{r_z}{\| \mathbf{r} \|}\right), \rho = f_{eq}\,\theta, x_u = \rho\,\frac{r_x}{\sqrt{r_x^2 + r_y^2}}, y_u = \rho\,\frac{r_y}{\sqrt{r_x^2 + r_y^2}} \quad (2)$$

  where $f_{eq}$is the equidistant focal parameter. OpenCV's fisheye coefficients handle distortion. (k_1, k_2, k_3, k_4)during calibration/undistortion.

- Extrinsics:
  The camera is rigidly attached to the PV frame. We define a tracker frame $\mathcal{T}$with $y_{\mathcal{T}}$as the rotation axis (single-axis tilt), $z_{\mathcal{T}}$normal to the module plane at $\beta = 0°$, and $x_{\mathcal{T}}$completing a right-handed triad. A one-time alignment step (Section 3.3.4) resolves the small offset between the camera optical axis and $z_{\mathcal{T}}$.

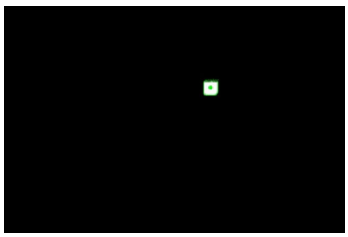


Figure 7. Compute the centre and draw the contour

Figure 8. Result of region segmentation based on the highest pixel intensity

2.3.2. Pixel-to-Ray Conversion

For each captured image, we first undistort the detected centroid. (u_s, v_s) using K, to obtain normalized coordinates (x_n, y_n). We then invert the fisheye projection to recover the unit view ray r_c in the camera frame:

$$\rho = \sqrt{x_n^2 + y_n^2}, \theta = \rho / f_{eq}, \mathbf{r}_c = \begin{bmatrix} \sin\theta \frac{x_n}{\rho} \\ \sin\theta \frac{y_n}{\rho} \\ \cos\theta \end{bmatrix}. \quad (3)$$

The ray is rotated into the tracker frame by $r_{\mathcal{T}} = R_{\mathcal{T}\leftarrow c}\, r_c$. Only the elevation component relative to the module's rotation plane matters for single-axis control. With $y_{\mathcal{T}}$ as the hinge axis, the instantaneous solar elevation in the module plane is

$$\alpha = \operatorname{atan2}(r_{z,\mathcal{T}}, r_{x,\mathcal{T}}), \quad (4)$$

where $r_{x,\mathcal{T}}$ and $r_{z,\mathcal{T}}$ are the x- and z-components of $r_{\mathcal{T}}$. α represents the desired module tilt to place the module normal toward the sun in the plane of rotation.

2.3.3. Commanded Tilt and Actuator Mapping

The commanded tilt is determined as

$$\beta = \alpha + \delta_0, \quad (5)$$

where δ_0 is a small fixed bias (from alignment calibration) that compensates camera–module boresight misalignment. The linear actuator converts β to stroke length L via the geometry of its mounting points (link length a, base separation b, anchor offsets). For a standard push-rod configuration,

$$L(\beta) = \sqrt{a^2 + b^2 - 2ab\cos(\beta + \phi)}, \quad (6)$$

with ϕ the mechanical offset at $\beta = 0°$. The motor driver then drives the 12 V actuator (25 W, 3 A; 10 mm/s) to the stroke corresponding to β, respecting the enforced limits (min–max angles) documented in Section 3.2.

2.3.4. Alignment and One-Time Calibration Procedure

The alignment and calibration procedure begins with intrinsic and fisheye calibration, in which a checkerboard sequence is captured to estimate the intrinsic matrix K, the fisheye distortion coefficients d, and the equidistant focal parameter. f_{eq}. Following this, extrinsic alignment is performed by placing the PV module at two known tilt angles. β_1 and β_2 (obtained from the gyroscope) under clear-sky conditions near solar noon so that the sun remains within the camera's field of view; for each pose, the corresponding image centroid (u_s, v_s) is recorded, the elevation angles α_1 and α_2 are computed as described in Section 3.3.2, and a least-squares optimization is solved to obtain the camera-to-tracker rotation matrix $R_{T\leftarrow c}$ and the alignment offset δ_0, which compensates for small mounting errors. The actuator geometry is then calibrated by measuring the actuator stroke length L at several known tilt angles β, enabling the estimation of a, b, and ϕ in the kinematic model $L(\beta)$, after which the inverse mapping L^{-1} is stored for real-time operation. During runtime, each incoming frame provides (u_s, v_s), which is converted into α and subsequently into the commanded tilt β^*; the gyroscope supplies the actual tilt β, and the controller drives the actuator until $|\beta - \beta^*| < \varepsilon$. This closed-loop control enforces mechanical limits (e.g., 25°–140°) and a daily reset routine. Under clear sky conditions, the geometric mapping provides a stable and consistent tilt angle with an acceptable error threshold (≤15%), whereas under heavily clouded conditions, the detected solar region may become irregular, leading α to follow the apparent bright centroid rather than the true sun position, thereby reducing the accuracy. This issue motivates the hybrid and adaptive enhancements discussed in the conclusions.

3. Results and Discussion

This result will be divided into two sections: the sunny and cloudy cases, where each section will also be divided into the brightest region and contour-finding methods. To establish a ground-truth reference for comparison, the "actual sun position" used throughout this section was computed from the site's geodetic latitude/longitude and measurement time stamps using sun position software, which returns solar zenith and azimuth. We then converted zenith to elevation and, for single-axis evaluation, projected the elevation into the tracker's rotation plane using the geometric mapping described in Section 3.3 to obtain the reference tilt used in our error plots. The site coordinates correspond to the UMPSA Pekan installation reported in the dataset (3.5439° N, 103.4358° E).

3.1. Sunny Day Cases

For this case, the data will be collected and processed. The image taken for this case is for a full day on 9 May 2021 from 9.45 am to 4.30 pm. For this study, 28 images were taken by the camera every 15 minutes. The target is to get a clearer image of the sun without any cloud noise. This is to make sure the experiment for this sunny case is successful, and to experiment with the method. The image taken is shown in Figure 9. All images for the sunny-day experiment were captured at Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Pekan campus. These coordinates correspond to

the physical installation site of the PV module and camera system, ensuring consistent solar path geometry and environmental conditions throughout the data collection period.

The image taken is then prepared for image processing. The wide-angle lens with camera proof from this image works well on a sunny day. The image or data is important since it is an essential part of image processing. The key for the brightest region method is to find the brightest or maximum value of the pixel in the image, as shown in Figure 10. A specific technique is used in this method: the grayscale inversion, applying a Gaussian filter, finding the brightest region, and circling the centre of that region. Meanwhile, the key to the contour finding method is to find the contour region and draw the contour and the centre of the contour. A specific technique is used in this method: the grayscale inversion, then applying a Gaussian filter, binary thresholding, a morphological technique, and lastly, finding the contour and drawing the centre of that contour. The result of these two methods is shown below by taking from the data set every 1-hour images.

From the image processing result, the data taken out from the process is the pixel centre value and the tilt angle of the PV panel. The result shows that this method successfully finds the brightest pixel, but does not specifically circle the sun's position. Therefore, it will be compared with the actual value for every measured data point. The contour-finding method, as illustrated in Figure 11, has demonstrated a more robust ability to isolate the solar region even when slight sky variations are present. Using grayscale conversion, Gaussian filtering, binary thresholding, and morphological refinement, this method delineates the sun's region before computing its centroid as the estimated solar position. As shown in Figure 11, the extracted contour provides a clearer and more stable representation of the sun compared to simpler intensity-based approaches. Although minor contour irregularities may occur when thin clouds or glare partially obscure the disc, the centroid-based position estimation remains relatively consistent. To quantitatively assess the method's reliability, each derived pixel coordinate and tilt angle was systematically compared with the reference dataset across all captured images.

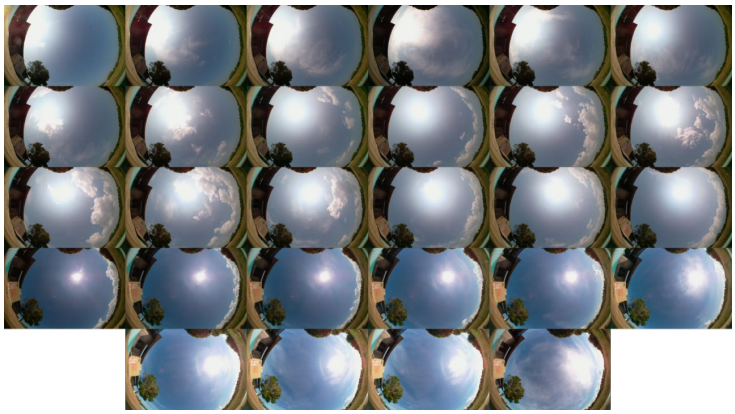


Figure 9. Sunny day data image

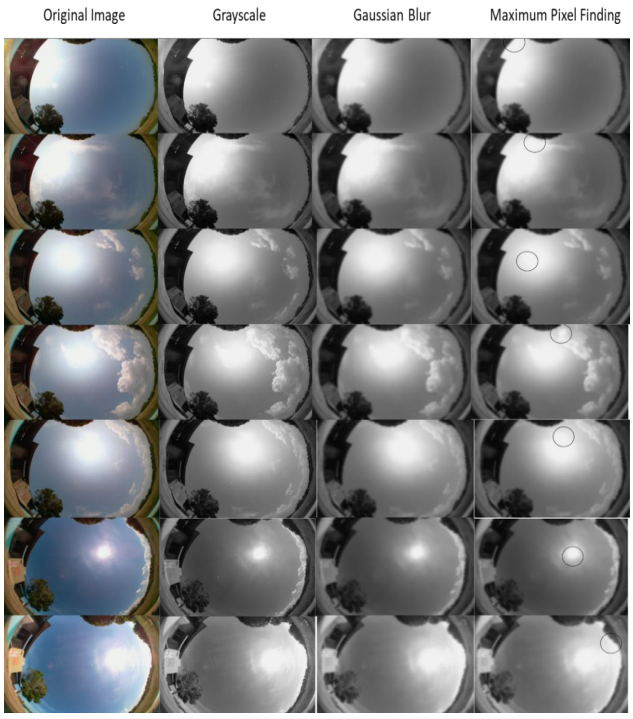


Figure 10. Result of the brightest region method

From the image processing result, the data taken out from the process is the pixel centre value and the tilt angle of the PV panel. The result shows that this method successfully finds the sun by looking at the result of contour findings. The thresholding method successfully finds the sun's position, but the shape is not a circle, as there is sometimes some cloud near the sun. For every measured data point, it will be compared with the actual value. As shown in Figure 12, the contour-finding method maintains a more coherent and continuous alignment with the actual pixel coordinates throughout the data-collection period. This consistency demonstrates the method's ability to provide stable centroid estimation despite minor variations in sky luminance. In contrast, the brightest-region method exhibits intermittent deviations from the reference curve, with several measurements scattered away from the actual trajectory. These fluctuations are primarily due to sensitivity to glare and reflective edges, which displace the maximum-intensity pixel from the true solar disc. Overall, Figure 12 highlights the superior tracking reliability of the contour-based approach for determining the sun's pixel position under clear-sky conditions.

The comparison of the actual and measured pixels shows that the contour finding method shows more data than the brightest region method. This is because the brightest region method data shown from the chart is scattered at a particular time. Nevertheless, the image processing image shows that although it is scattered, it still detects the sun's actual position in the image. Figure 13 compares pixel errors for both image-processing methods under sunny conditions. The contour-finding method demonstrates better detection, with most pixel values remaining within the error range that is acceptable. This indicates that the contour-based approach more effectively handles minor lighting variations and maintains proximity to the reference pixel path. In contrast, the brightest-region method displays a broader range of errors, particularly when reflective cloud edges or sky glare influence the image. These factors cause the brightest pixel to deviate from the actual sun position, resulting in less reliable measurements. Overall, Figure 13 demonstrates that the contour-finding method provides greater accuracy and stability in determining the sun's pixel position during clear-sky conditions.

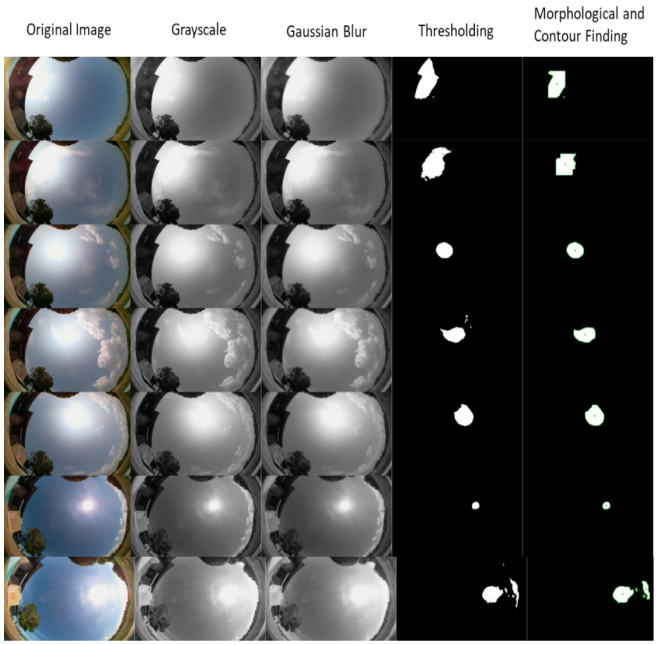


Figure 11. Result of contour finding method

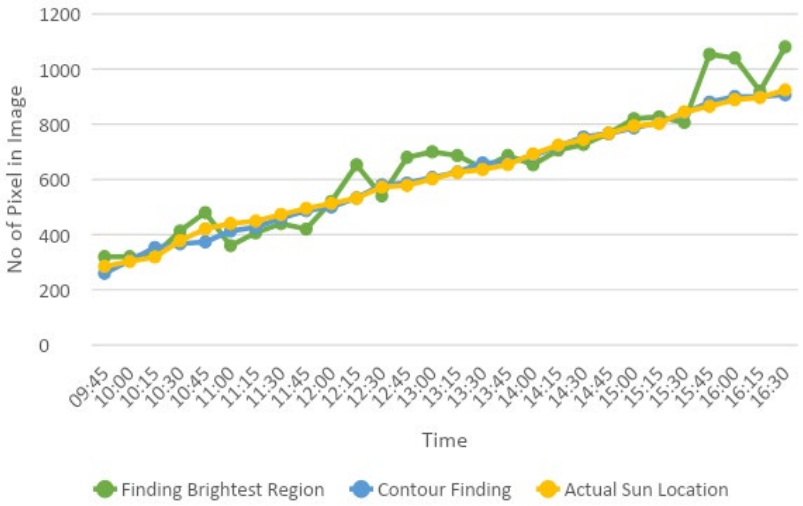


Figure 12. Pixel method comparison chart on 9 June 2021

Figure 14 compares tilt-angle errors from two image-processing methods used on sunny days. From the data shown, the error is divided into two. The measured pixel by applying the brightest pixel region is successful in detecting, but some data show an error above 10%. 18 errors did not exceed 10% of errors. The other 10 is an error above 10%. By applying the contour finding method, the measured pixel successfully detects most of the actual pixel positions. However, some of the data shows the error above 10%. 26 errors did not exceed 10% of the errors. The other 2 are errors above 10%.

Figure 15 compares tilt-angle errors for both image-processing methods under sunny conditions. From this comparison, the chart shows that the brightest region method is slightly scattered but close to the actual. From the chart, we can see that the value from around 12 pm to 1 pm is where the most errors occur. This is because there is a cloud nearby, and the detected brightest pixel is a reflection of sunlight from the cloud, although the cloud does not cover the sun. From the chart, we can see contour findings; most of the time, the measured tilt angle can match the actual tilt angle. This method is also almost accurate, as can be seen from the chart. Although it is slightly different at 1 pm, this method can detect it successfully. From the chart, the error is classified into two sections, above 15% and below 15%.

This method, in sunny cases, shows that it can detect the brightest region with slight error. This method can detect 71% with an error below 15%. In sunny cases, this method shows that it can detect the sun with a low number of errors. The contour finding method can detect 86% with an error below 15%. Overall, the contour-finding method accurately detected the sun's position in the image, with less than 15% error.

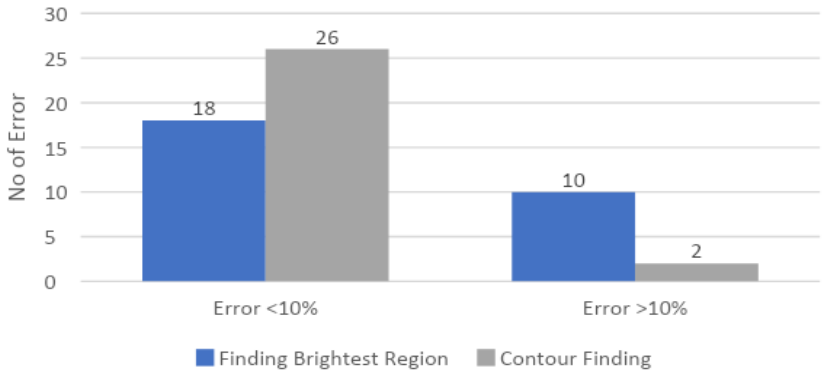


Figure 13. The pixel method compares the error chart

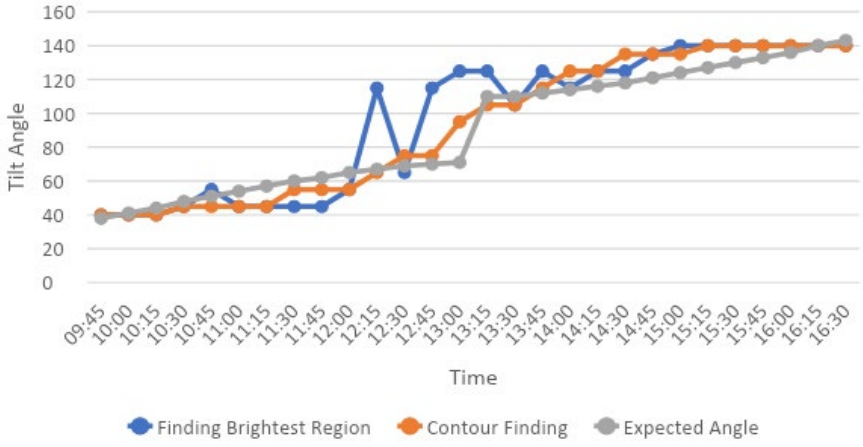


Figure 14. Tilt angle method comparison chart on 9 June 2021

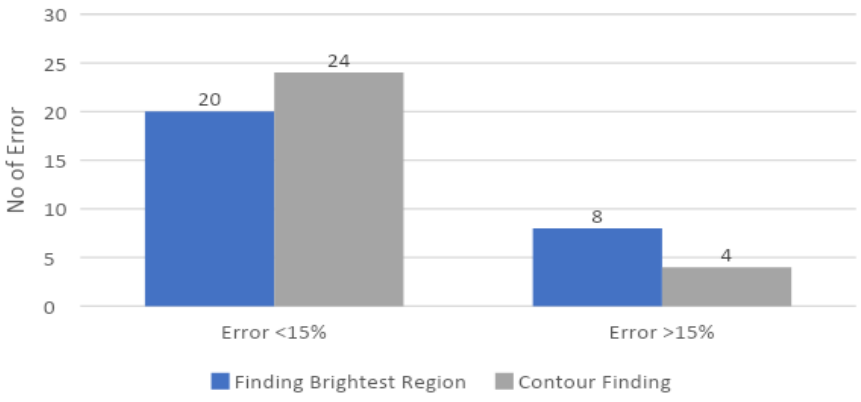


Figure 15. Tilt angle method comparison error chart

3.2. Cloudy Case

For this case, the data will be collected and processed. The image for this case was taken on 5 May 2021 from 10.54 am to 1.54 pm. The camera captured 11 images every 15 minutes. In this case, the target is to obtain a cloudy image of the sun and noise from the clouds. This is to ensure the experiment for this cloudy case is successful and to test the method. The sample image captured is shown in Figure 16.

The image taken is then prepared for image processing. The wide-angle lens with camera proof from this image works well on a cloudy day. The image or data is vital since it is an essential part of image processing. The key to the brightest region method is to find the brightest or maximum pixel value in the image. A particular technique is used in this method: the grayscale inversion, applying a Gaussian filter, finding the brightest region, and circling the centre of that region. Meanwhile, the key to the contour finding method is to find the contour region and draw the contour and the centre of the contour. A specific technique is used in this method: the grayscale inversion, then applying a Gaussian filter, binary thresholding, a morphological technique, and lastly, finding the contour and drawing the centre of that contour. These two methods are shown below by taking from the data set every 1-hour image. Figure 17 illustrates these processing steps. However, broken or dense clouds can produce bright edges or highlights that obscure the sun's disc, resulting in the detected maximum not always corresponding to the actual solar position, as demonstrated in Figure 17. In comparison, the contour-finding workflow under cloud cover applies grayscale inversion, Gaussian filtering, binary thresholding, morphological refinement, and centroid estimation. Figure 18 presents the outcomes of this method. Both sets of examples (Figures 17–18) utilize images captured at one-hour intervals to enable a direct, side-by-side comparison of each method's performance under identical weather conditions.

Figure 16. Cloudy day data image

Figure 17. Result of the brightest region method

From the image-processing results, the outputs include the estimated sun pixel-centre position and the corresponding PV module tilt angle. Under cloudy conditions, the contour-finding method demonstrates a stronger ability to identify the sun's approximate location despite the presence of significant cloud-induced noise. As illustrated in Figure 18, the thresholding and morphological operations successfully isolate a region corresponding to the bright solar area. However, the resulting contour may appear irregular due to diffuse illumination and cloud obstruction. Despite these challenges, the centroid of the detected contour provides a reasonable approximation of the sun's position. For each processed image, the measured pixel centre and derived tilt angle were subsequently compared against the actual reference values to evaluate the robustness of the contour-based approach under non-uniform sky conditions.

Figure 19 presents the comparison between the measured pixel-centre positions and the reference trajectory under cloudy-day conditions. From this image processing result, the data taken out from the process is the pixel centre value and the tilt angle of the PV panel. The result shows that this method successfully finds the sun by looking at the result of contour findings. The thresholding method successfully finds the sun's position, but the shape shown is too much noise as there is a cloud covering the sun. Therefore, it will be compared with the actual value for every measured data point. Accordingly, Figure 19 highlights the superior resilience of the contour-based approach for pixel-level localization in cloudy conditions.

Figure 20 presents the pixel-error comparison for both image-processing methods under cloudy-day conditions. As shown in the figure, comparing the actual and measured pixel shows that the contour finding method has shown that the value measured data is better and closer to the actual than the brightest region method. This is because the brightest region method data shown from the chart is scattered at a specific time. However, the result of the image processing shows that although it is scattered, it still detects the sun's actual position in the image.

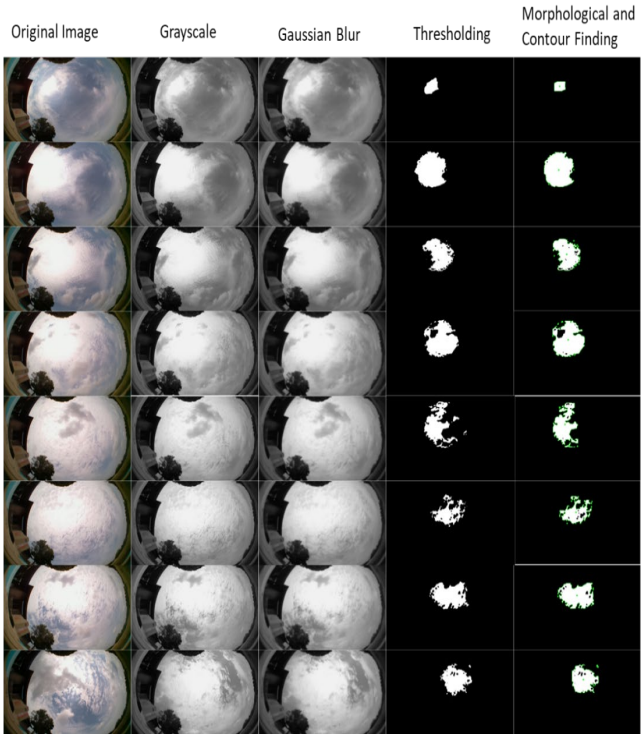


Figure 18. Result of contour finding method

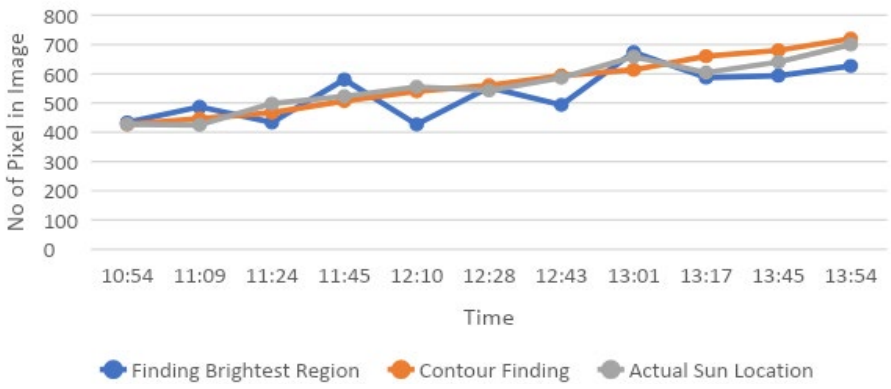


Figure 19. Pixel method comparison chart on 5 June 2021

Figure 21 summarises the tilt-angle comparison for both image-processing methods under cloudy-day conditions. As illustrated in the figure, the error is divided into two. The measured pixel by applying the brightest pixel region is unsuccessful in detecting most of the pixels, and there are more of the data show an error above 10%. 5 errors did not exceed 10% of the error. The other 6 are errors above 10%. Using the contour finding method, the measured pixel successfully detects all of the actual pixel positions with no data showing an error above 10%. 11 errors did not exceed 10% of error. There is zero error above 10%. Figure 22 compares tilt-angle errors for the brightest-region and contour-finding methods under cloudy conditions. Both methods show significant scatter from the reference tilt, indicating reduced reliability when diffuse irradiance obscures the solar centroid. From the tilt angle comparison chart, we can see that both methods show scattered data values, and most of the values are far from the actual value. This can be explained since this is for a cloudy case, the cloud covered around the sun is big, and since they reflect the light from the sun, the bright region got too big, or the contour object draws with a big area and much noise. So the tilt angle shows the found area's result, but it is far from the actual tilt angle, still detecting the bright area. Under these conditions, the brightest-region method achieves only 27% of detections within the ≤15% error criterion, while the contour-finding method reaches 63%. These results indicate that diffuse irradiance and cloud-edge glare often mislead the intensity-based detector, whereas region-based segmentation with centroid estimation is more resilient. However, both methods show reduced performance in overcast skies, highlighting the difficulty of locating a well-defined solar centroid when the disc is obscured. While the contour-finding method performs better under clouds, neither approach consistently achieves low tilt-angle error in heavily overcast scenes. Overall, these findings show that contour-based segmentation is generally more accurate and stable than intensity-peak detection in both weather conditions. However, cloud-induced diffusion remains a key limitation for image-based single-axis tracking and requires further algorithmic improvements, such as adaptive thresholding or hybrid image and astronomical models.

The key to contour finding is determining the optimum threshold value to generate the binary image. It is not easy to determine the optimal threshold value, and it should be set manually. Even under some interference, there is no optimum value to generate a circular object in a binary image, and some deviations from the actual sun centre are observed. For the brightest region method, the threshold value is excluded. It has no problem detecting the centre, but when a large or thick cloud blocks the view, the bright side of the cloud will appear, and the evaluated sun centre may be far from the actual sun centre.

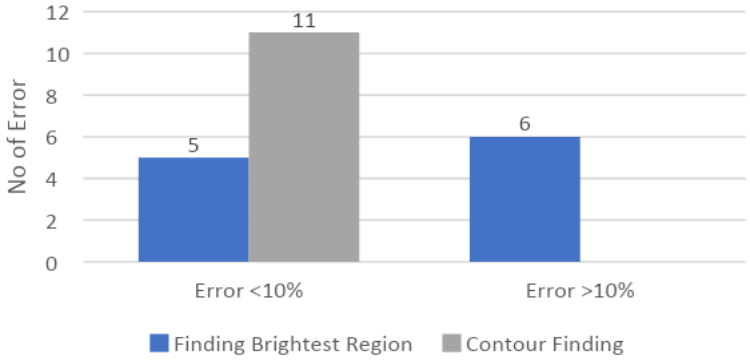


Figure 20. Pixel method compare error chart

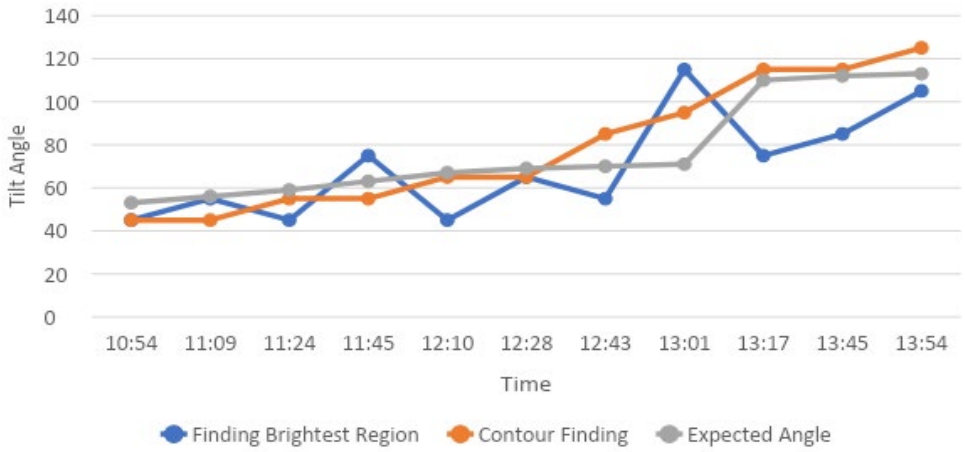


Figure 21. Tilt angle method comparison chart on 5 June 2021

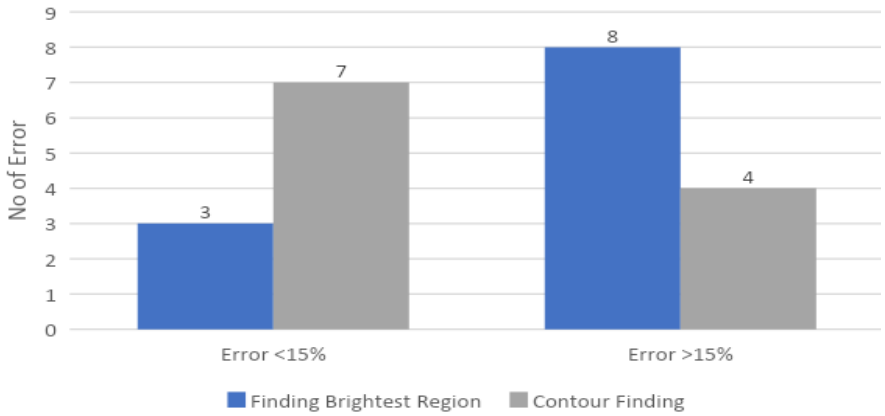


Figure 22. Tilt angle method comparison error chart

4. Conclusions

This paper proposes a single-axis solar tracker that will be based on the image processing method using OpenCV. The completion of this work comes from several levels that start from choosing the proper technique to get the precise tilt angle from the sun's position. First, several image processing techniques have been tested and discussed on several image datasets. Then, the acquisition of the tilt angle is accompanied by the image's pre-processing and processing of the value to get the tilt angle. Finally, the image processing has been compared and decided which is better for the system proposed. Based on the result obtained, it has been shown that this work's objective is to analyse and validate the performance of the proposed solar tracker and design a solar tracker that uses a single-axis tracking system based on the image analysis method, and extract the tilt angle from the image. The other objective is to develop a system that uses a linear actuator to operate according to the output from the system to make the PV panel move to the correct angle toward the sun's position. Based on the results obtained, the best image processing method has been obtaining to know the sun's precise location and using the actuator to move the PV panel to the desired angle from the image processing method. In this work, a real-time single-axis solar tracker using an image processing technique with high accuracy, reliability, and a wide-angle image that can detect a 180-degree angle of the picture was designed and applied to the system. Although there are some difficulties in detecting cloudy days, they can be overcome by adjusting the value in several methods and techniques used in the image processing method. In the recommendation to improve this work in the future, it is suggested that the sensor can be upgraded, like the tilt angle sensor, to get the reading of the tilt angle more precise during the operation. In terms of the camera, it can also be upgraded so that the quality of the picture will be better, and it can use a filter to help reduce the noise. The image processing method can also be improved in many ways, such as methods that have not been discovered. On the other hand, OpenCV is still upgrading the method over time to overcome such weaknesses. Then the method will be upgraded over time.

Acknowledgement

The researchers would like to acknowledge Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA) and Muhammadiyah Semarang (UNIMUS) for supporting this project.

Funding

This work was supported by Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA) and Muhammadiyah Semarang (UNIMUS) for their financial support and facilities through the International Research Matching Grant (University reference RDU242729/UIC241531).

Declaration of Competing Interest

The authors declare no conflict of interest.

CRediT Authorship Contribution Statement

Mohd Shawal Jadin (Conceptualisation; Methodology; Writing – original draft; Writing – review & editing)
Sabhan Kanata (Methodology; Resources; Funding acquisition)
Mohd Herwan Sulaiman (Validation; Writing – review & editing)
Mohd Redzuan Ahmad (Formal analysis; Writing – review & proofreading)
Kamarulazhar Daud (Writing – review & editing)

Availability of the Data and Materials

The data used to support the findings of this study are included within the article.

Ethical Declaration

This research did not involve any human participants, animals, or sensitive personal data. Therefore, ethical approval was not required. All data used in this study were obtained from publicly available sources and used in accordance with relevant guidelines and regulations.

Generative Artificial Intelligence Declarations

Artificial intelligence tools were used to assist in drafting sections of the manuscript and summarizing literature. The authors reviewed and verified all AI-generated content to ensure accuracy and originality. Final interpretations and conclusions are the responsibility of the authors.

REFERENCES

[1] G. Di Foggia, M. Beccarello, and B. Jammeh, "A global perspective on renewable energy implementation: Commitment requires action," *Energies*, vol. 17, no. 20, p. 5058, Oct. 2024, doi: 10.3390/en17205058.
[2] M. R. Nagaraja, W. K. Biswas, and C. P. Selvan, "Advancements and challenges in solar photovoltaic technologies: Enhancing technical performance for sustainable clean energy—A review," *Solar Energy Adv.*, vol. 5, p. 100084, 2025, doi: 10.1016/j.seadv.2024.100084.
[3] K. Sezen, "Comparative analysis of east–west and south–north single-axis solar tracking systems," *Konya J. Eng. Sci.*, vol. 13, no. 2, pp. 624–641, May 2025, doi: 10.36306/konjes.1584923.
[4] R. Sadeghi, M. Parenti, S. Memme, M. Fossa, and S. Morchio, "A review and comparative analysis of solar tracking systems," *Energies*, vol. 18, no. 10, p. 2553, May 2025, doi: 10.3390/en18102553.
[5] H. Jafari, A. Poursalan, A. Gholami, R. Gavagsaz-Ghoachani, and M. Phattanasak, "A review of solar tracking technologies: Mechanisms, challenges, and future directions," in *Proc. Int. Conf. Materials & Energy (ICOME-EE)*, Bangkok, Thailand, 2024, pp. 1–6.
[6] N. E. Herrera-Carrillo, M. Rivero, E. Gomez-Ramirez, and R. Loera-Palomo, "Solar irradiance estimation based on image analysis," in *Proc. IEEE Int. Autumn Meeting Power, Electron. Comput. (ROPEC)*, Ixtapa, Mexico, 2018, pp. 1–5, doi: 10.1109/ROPEC.2018.8661413.
[7] M. E. H. Chowdhury, A. Khandakar, B. Hossain, and R. Abouhasera, "A low-cost closed-loop solar tracking system based on the sun position algorithm," *J. Sens.*, vol. 2019, p. 3681031, Feb. 2019, doi: 10.1155/2019/3681031.
[8] A. H. Musa, E. C. Alozie, S. A. Suleiman, J. A. Ojo, and A. L. Imoize, "A review of time-based solar photovoltaic tracking systems," *Information*, vol. 14, no. 4, p. 211, Mar. 2023, doi: 10.3390/info14040211.
[9] N. Jannah, T. S. Gunawan, S. H. Yusoff, M. S. Abu Hanifah, and S. N. M. Sapihie, "Recent advances and future challenges of solar power generation forecasting," *IEEE Access*, vol. 12, pp. 168904–168924, Nov. 2024, doi: 10.1109/ACCESS.2024.3494558.
[10] J. Gaboitaolelwe, A. M. Zungeru, A. Yahya, C. K. Lebekwe, D. N. Vinod, and A. O. Salau, "Machine learning based solar photovoltaic power forecasting: A review and comparison," *IEEE Access*, vol. 11, pp. 40820–40845, Apr. 2023, doi: 10.1109/ACCESS.2023.3268571.
[11] A. L. B. José, R. M. Varghese, R. Mathew, and P. V. Sujith, "Dual-axis solar tracking system with LDR sensors," *IET Conf. Proc.*, vol. 2023, no. 6, pp. 112–117, May 2023, doi: 10.1049/icp.2023.1234.
[12] K. B. de Melo, H. S. Moreira, and M. G. Villalva, "Influence of solar position calculation methods applied to horizontal single-axis solar trackers on energy generation," *Energies*, vol. 13, no. 15, p. 3826, Jul. 2020, doi: 10.3390/en13153826.
[13] A.-H. I. Mourad, H. Shareef, N. A. Ameen, A. H. Alhammadi, M. Iratni, and A. S. Alkaabi, "A state-of-the-art review: Solar trackers," in *Proc. Advances Science Eng. Technology Int. Confs. (ASET)*, Dubai, UAE, 2022, pp. 1–6, doi: 10.1109/ASET53987.2022.9734898.
[14] N. Rajkumar, G. Jeyasri, A. Kumar, and A. Mahilarasi, "Solar tracking methods: A comprehensive survey," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 4, pp. 1432–1439, Apr. 2024, doi: 10.22214/ijraset.2024.60112.
[15] M. Hammas, H. Fituri, A. Shour, A. A. Khan, U. A. Khan, and S. Ahmed, "A hybrid dual-axis solar tracking system: Combining light-sensing and time-based GPS for optimal energy efficiency," *Energies*, vol. 18, no. 1, p. 217, Jan. 2025, doi: 10.3390/en18010217.
[16] C. D. Rodriguez-Gallegos, O. Gandhi, S. K. Panda, and T. Reindl, "On the PV tracker performance: Tracking the sun versus tracking the best orientation," *IEEE J. Photovolt.*, vol. 10, no. 5, pp. 1474–1480, Jul. 2020, doi: 10.1109/JPHOTOV.2020.3001647.
[17] A. Niccolai and A. Nespoli, "Solar position identification on sky images for photovoltaic nowcasting applications," in *Proc. IEEE Int. Conf. Environment Elect. Eng. & IEEE Industrial Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, Madrid, Spain, 2020, pp. 1–5, doi: 10.1109/EEEIC/ICPSEurope49358.2020.9160539.
[18] J. Rishmany, C. Lahoud, J. Harmouche, R. Imad, and N. Saba, "Advancements in solar tracking: A comprehensive review of image-processing techniques," *Sustainability*, vol. 18, no. 2, p. 1117, Jan. 2026, doi: 10.3390/su18021117.