Image-based Solar Tracker Using Raspberry Pi

R Abd Rahim, M.N.S.Zainudin, M.M.Ismail, M.A.Othman
Centre for Machine Learning for Signal Processing
Fakulti Kej. Elektronik dan Kej. Komputer (FKEKK),
Universiti Teknikal Malaysia Melaka (UTeM)
rosman@utem.edu.my, noorazlan@utem.edu.my, muzafar@utem.edu.my, azlishah@utem.edu.my

***Abstract*— Malaysia is a country that receives the sun light throughout the year. Sun light can be used as an alternative energy to fossil-fuel or hydroelectricity station to generate electricity. There are various methods used to optimize the harvesting of solar energy but some were costly while others often cannot give precise location of the sun especially when there is less illumination from the sun. This project present a solar tracker system that uses a webcam as the main sensor combined with image processing technique that is embedded into Raspberry Pi board to locate the location of the sun in the sky. Raspberry Pi is the main board used replacing the big computer CPU to process the image. Two motor servo with design of pan and tilt were used to move the webcam to follow the direction the sun. It was shown that the system manage to catch the position of the sun even during the cloudy day. The system offers simple implementation of the tracker with the capability to locate the central coordinate of the sun on an image and the Raspberry Pi will send signal to the motor that will rotate in accordance with the movement of the sun**

Keywords—Solar energy; renewable energy; solar tracker; image processing; raspberry-pi

I. Introduction

Solar is a renewable energy and can be used continuously and consistently, mostly in the countries that receive sun light throughout the year. It is predicted that the cost for energy could be reduced in the sunbelt countries by 2020 if solar energy is widely used [1]. In order to harvest solar energy efficiently, solar tracker can be used to ensure maximum harvesting possible as it can track the position of the sun and at the same time capture the radiation at maximum level by facing the solar panel directly to the sun.

Some tracker in the current implementation and researches uses sensors like photodiode and phototransistor [2]-[4]. These types of sensors locate location of the sun by sensing barely on the sunlight intensity. The disadvantages of this type of tracker is that it has a high sensitivity to weather condition particularly on temperature and humidity change as well as rapid deterioration under extreme condition [5],[6]. There are other type of solar tracker that uses complex control system and circuitry [7], though it overcomes the high sensitivity problem, it incurs comparatively high maintenance cost.

In order to solve the problems mentioned above, an image-based sun position sensor has been developed in recent years. High precision solar system by using low cost webcam was developed by Minor *et. al* [8] that was able to locate the sun and extrapolate its position when it cannot be observed for a period of time with tracking accuracy of 0.1° without being affected by weather condition. Another image-based sun position sensor and a tracking controller with image processing algorithm [9] established a sun image tracking platform that is capable to addressed the problem of unstable tracking in cloudy weather and achieve a tracking accuracy of 0.04°. other works on tracking the sun using the image of the sun [5],[10],[11] prove the possibility of tracking the sun by using image processing technique and these motivate the research presented in this paper.

This paper present a solar tracker system by using digital image processing algorithm as the core element and the cost can be reduced by using a webcam as an alternative to a high cost camera. The image processing will be embedded on a board named Raspberry Pi. The Raspberry Pi is a credit-card-sized single-board computer developed in the United Kingdom by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. There are two model which both model are similar except for model B have the Ethernet, 2 USB ports and 512 MB SDRAM. Both models can run a Linux operating system. The model that use in this project is model B.

II. Methodology

A. *Hardware Design and Operation*

The hardware design includes the Raspberry Pi, two servo motors in pan and tilt position and the webcam. The Raspberry Pi as the main board that processes image and will control the servo motor. The webcam will capture the image from the sky and the will send it to the Raspberry Pi through Universal Serial Bus (USB). The servo motor connected to Raspberry Pi using GPIO port. After locating the position of the sun, Raspberry Pi will move the servo motor, pan or tilt which one are necessary or both so that the sun will positioned at the centre of the image.

If no sun is detected in the image, the Raspberry Pi will move the servo motor until the sun is found. If the sun already at the centre of the image, the Raspberry Pi will give signal to the servo motor to stay at that position for 10 minute.

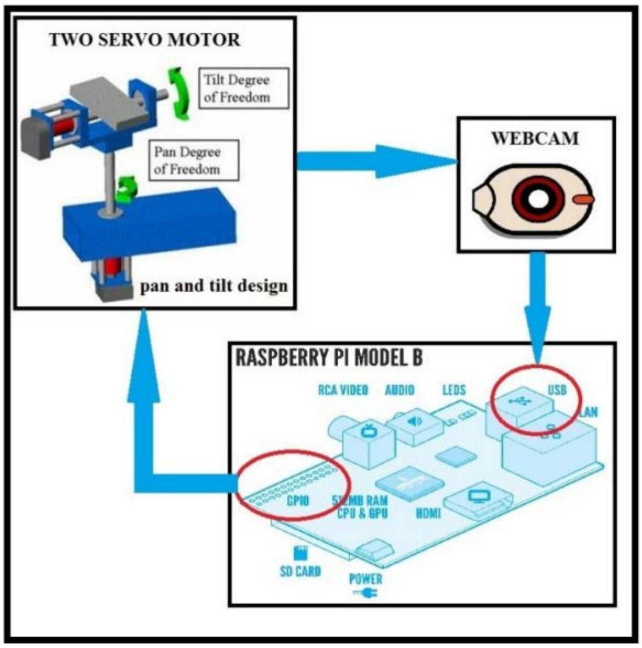


Fig. 1. The solar tracker design

B. *Image Processing*

Since the image taken from the webcam was a 24-bit colour image, it was converted to 8-bit grayscale colour to allow efficient processing.

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

This process used OpenCV module. Frame is the source image. cv2.COLOR_BGR2GRAY is the colour code that provided by OpenCV. BGR is blue, green and red. In OpenCV the bytes of image are reversed so it starts from blue, green then red. In this coding, it convert 24-bit colour image into 8 bit colour image.

The grayscale image was then being converted to binary image in order to detect the circular or curve shape of the sun.

thresh = cv2.adaptiveThreshold((src, max Value, adaptive Method, threshold Type, block Size, C)

Adaptive Threshold is the special method to convert the grayscale image into the binary image. The first parameter in adaptive Threshold is the source image. The max Value is a non-zero value assigned to the pixels for which the condition is satisfied.

Adaptive method is the adaptive thresholding algorithm to use. There only have two types of adaptive method, there are ADAPTIVE_THRESH_MEAN_C or ADAPTIVE_THRESH_GAUSSIA_C. Threshold Type only has two types either THRESH_BINARY or THRESH_BINARY_INV.

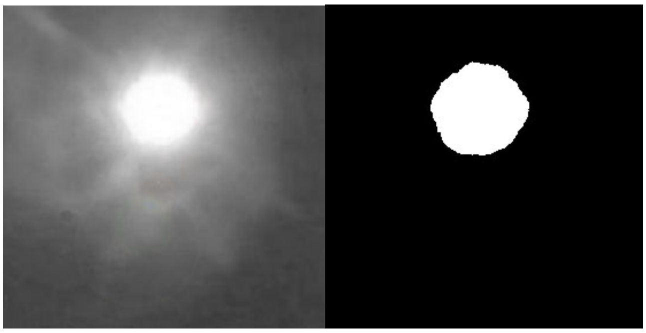
THRESH_BINARY

$$dst(x,y) = \begin{cases} maxvalue & \text{if } src(x,y) > T(x,y) \\ 0 & \text{otherwise} \end{cases}$$

THRESH_BINARY_INV

$$dst(x,y) = \begin{cases} 0 & \text{if } src(x,y) > T(x,y) \\ maxvalue & \text{otherwise} \end{cases}$$

Block Size is size of a pixel that is used to calculate the threshold value for the pixel (3, 5, 7 and so on). C is the constant subtracted use in adaptive method. It can be any number positive, zero or negative.



(a) (b)

Fig. 2. (a) The Grayscale image of the sun. (b) The binary image of the sun.

To identify the shape of sun, the Hough transformation is use. Circle shape is use because the shape of sun is circle.

cv2.HoughCircles(image, method, dp, minDist[, param1[, param2[, minRadius[, maxRadius]]]])

Image is the source image; it can be in 8-bit image or binary image. *Method* is the detection method being used, in this project cv2.CV_HOUGH_GRADIEN is used. *dp* is an inverse ratio of the accumulator resolution image resolution. If *dp* is one, the accumulator will have same resolution with the source image. If *dp* is two the accumulator resolution will be half of source image.

Accumulator resolution:

$$Accumulator\ resolution = \frac{source\ image\ resolution}{dp}$$

Param1 and *param2* is the first and second method-specific parameter. minRadius is the minimum circle radius used in detection. maxRadius is the maximum circle radius will be use in detection.

C. *Motor Control*

The movement of servo motor was determined by the Pulse Width Modulation (PWM) signal send from Raspberry Pi. The PWM signals were controlled by the position of the sun in the image. Figure 3 shows the flowchart that analyzes the process of determining the coordinate of the sun by first detecting the shape of sun.

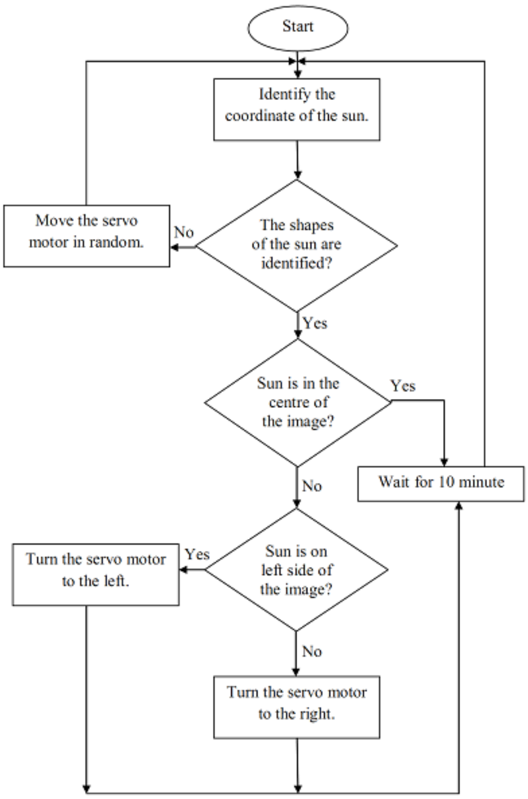


Fig. 3. The flow of motor servo operation.

If the sun is not detected on the image due to the webcam facing directly to the sun, Raspberry Pi will send a random acceptant PWM signal to servo motor. This will continuous until the sun is in the image. When the sun is detected, the Raspberry Pi will then analyse the position of the sun. This applies for both pan and tilt motors. If the sun is at the left or right of pan motor, Raspberry Pi will send signal to pan motor signal so that it will position the sun at the centre.

Figure 4 shows the pan and tilt control line image with sun on the left side. Raspberry will give PWM signal to move the pan servo motor so that the pan line comes across the centre of the sun. The same processes apply to tilt servo motor.

After the sun is at the centre of the image, the Raspberry Pi will continuously send the same PWM signal for 10 minute. This is a closed-loop system application where the output will determine the input.

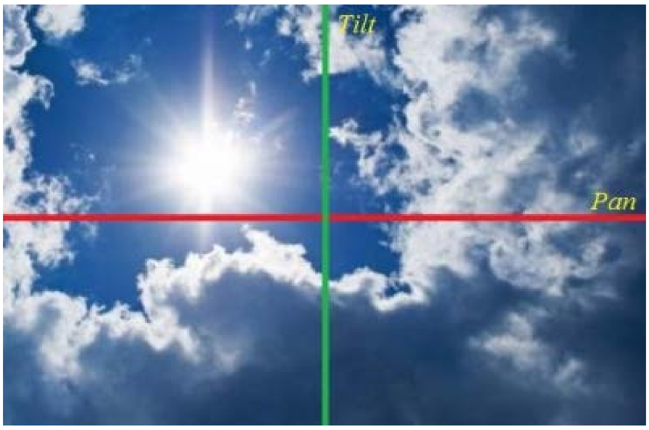


Fig. 4. Pan and tilt control.

D. *PWM Control*

There were two methods being used in controlling PWM in servo motor, one is by using GPIO control and another is by using PWM library. The time sleep needs to be controlled so that the system can send PWM signal to the motor servo. In GPIO.output (pin, True) it will give high signal or one.

Referring to Figure 5, Time sleep is the delay that refers to the time that the true signal will be continuously sent to servo motor. Time *a* is from 1ms to 2ms based on the degree of angle needed to make the webcam facing to the sun.

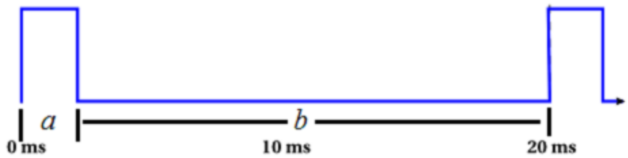


Fig. 5. The relation between time a and b.

The relation between time *a* and *b* is as follow:

$b = 20ms - a$

III. Results and Performance

A. *Hardware*

The PWM signal of pan and tilt servo motor are connected to GPIO port 22 and GPIO port 23. There is not output power for servo motor which both servo motor are powered by Raspberry. The webcam are directly connected to USB port 1.

Fig. 6. The complete hardware in this project

B. *Image Processing*

The webcam support 2048 x 1536 pixels and 3.15 MP images. It also comes with autofocus and dual-LED flash.

Fig. 7. The original image from webcam

Figure 7 is an image taken from the webcam during the noon with some clouds one hour after rain. The shape of the sun is not very clear because of the cloud. Figure 8 shows similar image with image processing applied. The orange circle is showing that the location of the sun is detected and the orange dot inside the circle is the centre of the sun.

The red line is the pan servo motor and the green line is tilt servo motor. It was assumed that both servo motor were at 90 degree. When the location of the centre of the sun is located, the servo motor will move the webcam to the centre of the sun. The cross point between pan line and tilt line is the centre of the image.

The pan servo motor is needed to reduce the degree of the rotation so that the pan line will cross the orange dot which is the centre of the sun. At 90 degree, the high signal PWM signal send from Raspberry Pi was measured to be 1.47ms as indicated in Table 1 (image 3). In accordance to that when the PWM signal sent to servo motor is lower, which is at 1.26ms the degree of rotation will be at 70 degree.

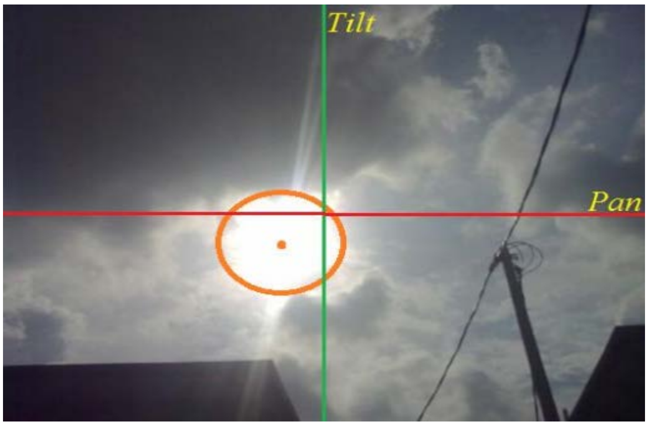


Fig. 8. Image analysis

The same methods were applied to tilt motor servo. The tilt line needs to cross the orange circle. Since the orange circle is at the left side the degree rotation of the tilt servo motor need to be reduced. As the tilt line are at 90 degree so it will be reduced to 75 degree and signal sent took 1.32ms. Data for different images were recorded as in Table 1.

The end result is shown in Picture 4.15 where the centre of the image is at the centre of the sun as assume that the cross between pan and tilt line are the centre of the image.

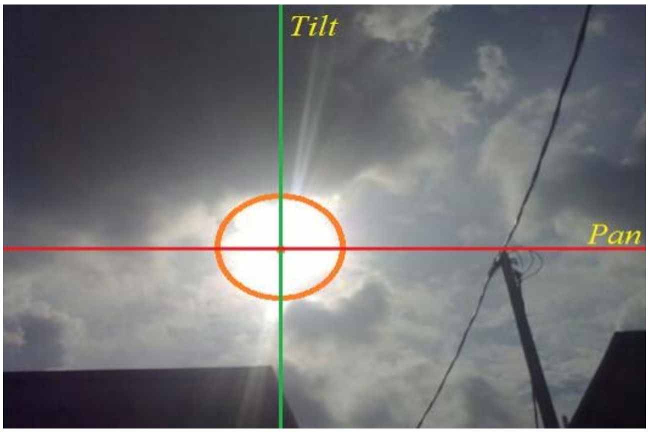


Fig. 9. Location of the sun

TABLE I. Time *a* and *b* with respect to degree of the motor rotation for different image.

a	b	Degree of rotation	Image
0.54ms	19.46 ms	0	Image 1
1.01 ms	18.99 ms	45	Image 2
1.47 ms	18.53 ms	90	Image 3
1.94 ms	18.06 ms	135	Image 4
2.40 ms	17.60 ms	180	Image 5

Referring to Table 1, *a* is the high signal of the PWM sending to servo motor that determine the degree of the rotation of the motor.

IV. Conclusion

From this work, an image-based solar tracking system using Raspberry Pi was proposed. It was shown that the system works very well with the webcam aided with image processing embedded in Raspberry Pi board always manage to catch the position of the sun even during the cloudy day. This will certainly result in maximum harvesting of solar energy. The system offers simple implementation of the tracker with the capability to locate the central coordinate of the sun on an image and the Raspberry Pi will send signal to the motor that will rotate accordingly follow the movement of the sun.

The proposed tracking system also does not present the disadvantages of using photodiode or phototransistor that deteriote over time and less efficient in a changing weather environment. For future work, the system needs to be optimised by making appropriate adjustment and improvement to the algorithm to suite the real-time operation on the field.

REFERENCES

[1] Hauff, J.; Verdonck, M.; Derveaux, H.; Dumarest, L.; Gammal, A.E.; Llamas, P.; Masson, G. "*Unlocking the Sunbelt Potential of Photovoltaics*"; European Photovoltaic Industry Association and Others: Brussels, Belgium, 2010.

[2] W. A. Lynch and Z. M. Salameh, "*Simple electro-optically controlled dual-axis sun tracker*," Solar Energy, vol. 45, pp. 65-69, 1990.

[3] P. A. Davies, "*Sun-tracking mechanism using equatorial and ecliptic axes*," Solar Energy, vol. 50, pp. 487-489, 1993.

[4] S. A. Kalogirou, "*Design and construction of a one-axis sun-tracking system*," Solar Energy, vol. 57, pp. 465-469, 1996.

[5] Morteza Taki, Yahya Ajabshirchi, Hossein Behfar, Mohsen Taki, "*Experimental Investigation and Co nstruction of PV Solar Tracker Control System Using Image Processing*", Modern Applied Science, Canadian Center of Science and Education, Vol. 5, No. 6; December 2011.

[6] P. J. Hession and W. J. Bonwick, "*Experience with a Sun Tracker System*," Solar Energy, vol. 32, pp. 3-11, 1984.

[7] H. D. Maheshappa , et al., "*An improved maximum power point tracker using a step-up converter with current locked loop*," Renewable Energy, vol. 13, pp. 195-201, 1998.

[8] Minor M. Arturo, García P. Alejandro, "*High–Precision Solar Tracking System*", Proceedings of the World Congress on Engineering 2010 Vol II

[9] Cheng-Dar Lee, Hong-Cheng Huang and Hong-Yih Yeh, "*The Development of Sun-Tracking System Using Image Processing*", Sensors 2013, 13, 5448-5459; doi:10.3390/s130505448

[10] Omar BadranP, P and Ismail Arafat, "*The Enhancement of Solar Distillation using Image Processing and Neural Network Sun Tracking System*", International Journal of Mining, Metallurgy & Mechanical Engineering (IJMMME) Volume 1, Issue 3 (2013) ISSN 2320-4052; EISSN 2320-4060

[11] Rajwant Kaur, Sukhpreet Kaur, "*Object Extraction and Boundary Tracing Algorithms for Digital Image Processing: Comparative Analysis: A Review*", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013