Solar Tracking Using Image Processing in OpenCV

COSC 602

Team: Gretchen Hook-Podhorniak, Mark Shanks

Introduction

Traditionally, solar tracking is complicated by many variables, including the altitude and azimuth angles of the sun, which change based on the season, time of day, and latitude of the location in question. In order to absorb the most direct rays, solar tracking makes small adjustments so that the device is always perpendicular (or other specified angle) relative to the sun [1]. There are several different types of solar trackers in development which can operate either passively or actively, on a single or double axis, and with closed or open loop feedback mechanisms. Passive trackers operate without any mechanical techniques to direct them towards the most direct beams, while active trackers use a sensor to detect the position of the sun in the sky and adjust the position of the device [1]. Trackers can also operate on an open loop, which computes the position of the sun based on the current state of the sun and a predetermined algorithm, while closed loop systems use a feedback control loop that manipulates the input, and then uses that output to orient the tracker [1].

Problem Description

For this project our team focused on developing a system capable of using image processing techniques to identify and track the sun to redirecting its rays towards a shady spot in a yard, such as underneath a tree, which is unable to grow grass due to lack of direct sunlight. Redirecting sunlight to shady areas involves tracking the movement of the sun in the camera's view frame in order to send corresponding directing signals to a pan/tilt mechanism outfitted with a mirror. Our objective was to accomplish this using a camera/mirror system that could be attached to a roof to track the motion of the sun throughout the day and redirect its light downward to a fixed position in a yard below. Figure 1 shows a simplified conceptual diagram of the system in action.

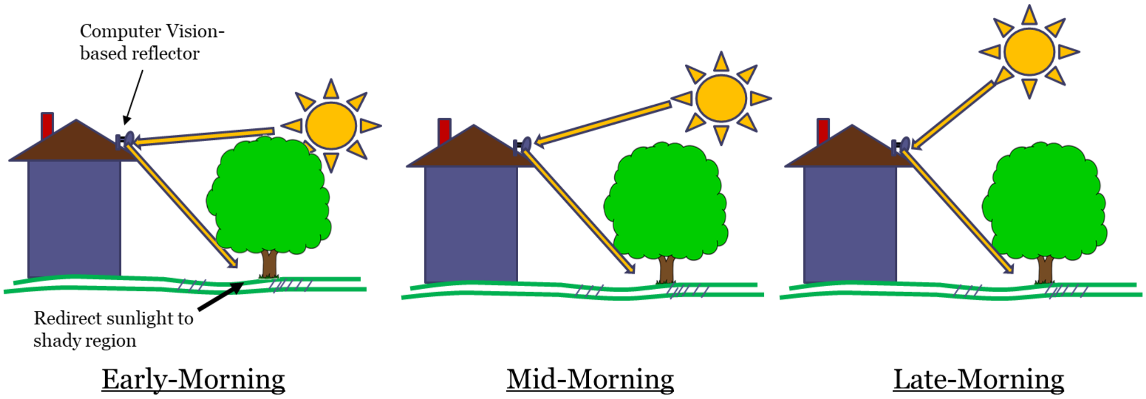

Fig. 1 Conceptual diagram of proposed system operating, maintaining reflection angle to desired location as the sun position changes with time.

Experiment Design

The proof of concept system uses standard off the shelf components and open source software. Table 1 includes a Bill of Materials, not including expendables (e.g. glue, nails, mounting platform, etc.). Figure 2 shows the experimental setup for the prototype components. OpenCV and Python development environments were installed on the Raspberry Pi with the aid of an online tutorial [2].

Sun Tracker: Bill of Materials	
Logitech Webcam (720 X 1280p)	$22
Adafruit Pan/Tile Servo Assembly	$30
Raspberry Pi 3b (keyboard, monitor, and mouse not included), with Raspbian Stretch OS.	$35
Connection Wires (10 pk)	$5
Small Mirror	$3
Software: OpenCV 3	$0
Software: Python 2.7	$0

Table 1 Prototype System Bill of Materials

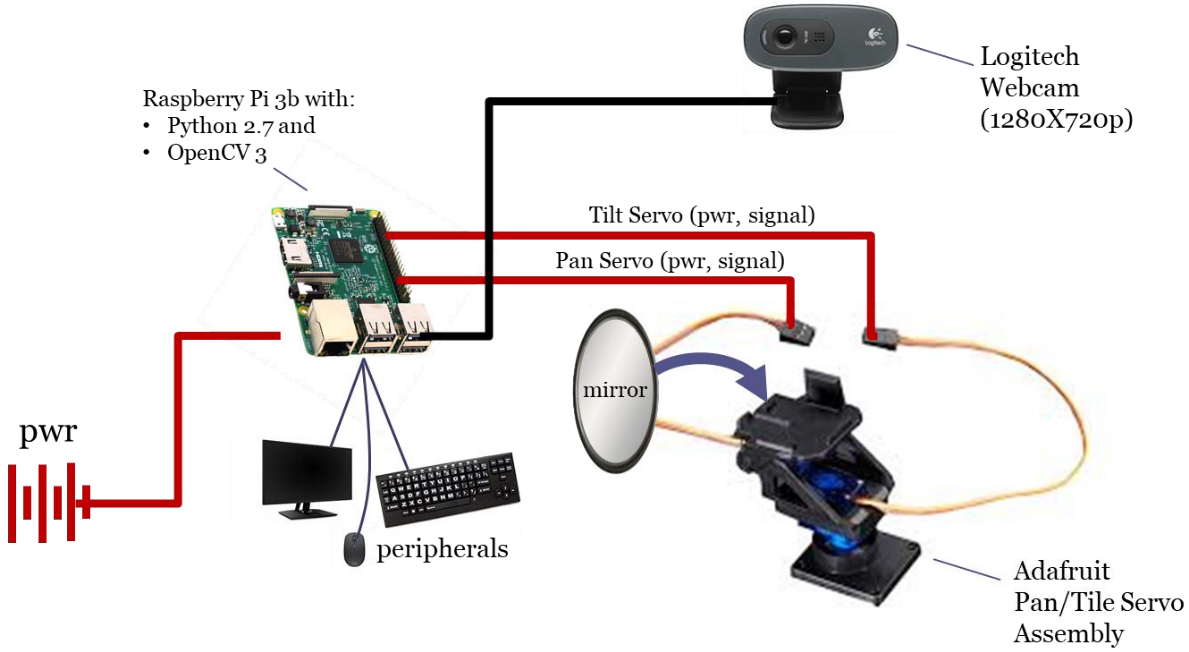

Fig. 2 Experimental prototype schematic.

Our tracker uses a standard webcam to sense the sun within the camera's field of view (FOV). To determine the horizontal and vertical angular FOV, both the lateral and vertical extents of the image frame were physically measured using a tape measure at a known distance from the camera and derived using simple trigonometry. Finally, these horizontal and vertical angles were divided by the cameras length and height resolution, respectively, to determine the degreed per pixel (DPP).

The software used an OpenCV function to obtain a live video feed at 30fps, and our image processing functions operated on each frame. This speedy operation helped to validate the system was tracking properly, however, given the slow movement of the sun in the sky, a much slower input frame rate (e.g. 1 per minute), could be used. The software converts an imput frame image to grayscale, after which Gaussian convolutional smoothing is used. The smoothing function helped to reduce the number of noise in the image. Next, a threshold function was applied. We assume that the largest and brightest object in the sky will be the sun, so following some threshold value tests, the selected threshold was set on the high end at 200.

From this threshold image the contours function (chain 8 connectivity) was used to locate the objects in the field of view. In OpenCV, a variety of object features can be derived from each contour (i.e. object) identified, by using image moments. These image moments were used to find the object areas, from which the largest area was selected – the largest/brightest object is assumed to the sun image object. Next, the centroid of the sun object was derived to obtain the sun's position in the image frame. We then use the centroid value to adjust the position of the mirror.

The centroid value, in conjunction with the pre-determined camera DPP value, the known position of the camera relative to the horizontal ground plane (earth), as well as the pre-determined intended illumination target, are used by the software to find the sun's actual position in the sky in degrees azimuth and elevation. Given that the mirror pan/tilt assembly axis of rotation is aligned with the central axis of the camera, the sun's elevation/azimuth position angles relative to the illumination target are divided by 2, which are the appropriate mirror pan/tilt position angles required to reflect the sunlight towards the target.

Experimental Results

Figure 3 shows the prototype experimental setup. The camera and pan/tilt mirror assembly were mounted to a mock-roof to ensure their bases were aligned to the same planar axis. A "+" sign was made on the wall using electrical tape, which represents the pre-determined illumination target. As stated, the objective is for the sun image to be able to move anywhere in the view frame while the reflected sunlight maintains position on the target.

Fig. 3 Prototype experimental setup.

Figure 4 shows photos from experimental testing. A flashlight was used in a moderately lit room to simulate the presence of the sun. Note that as an aid to ensure the tracking functions were performing properly, in the software, we included a solid black circle centered on the centroid of the sun’s image in the view frame. The black line tracing the sun image ‘corona’ is the contour of the sun image.

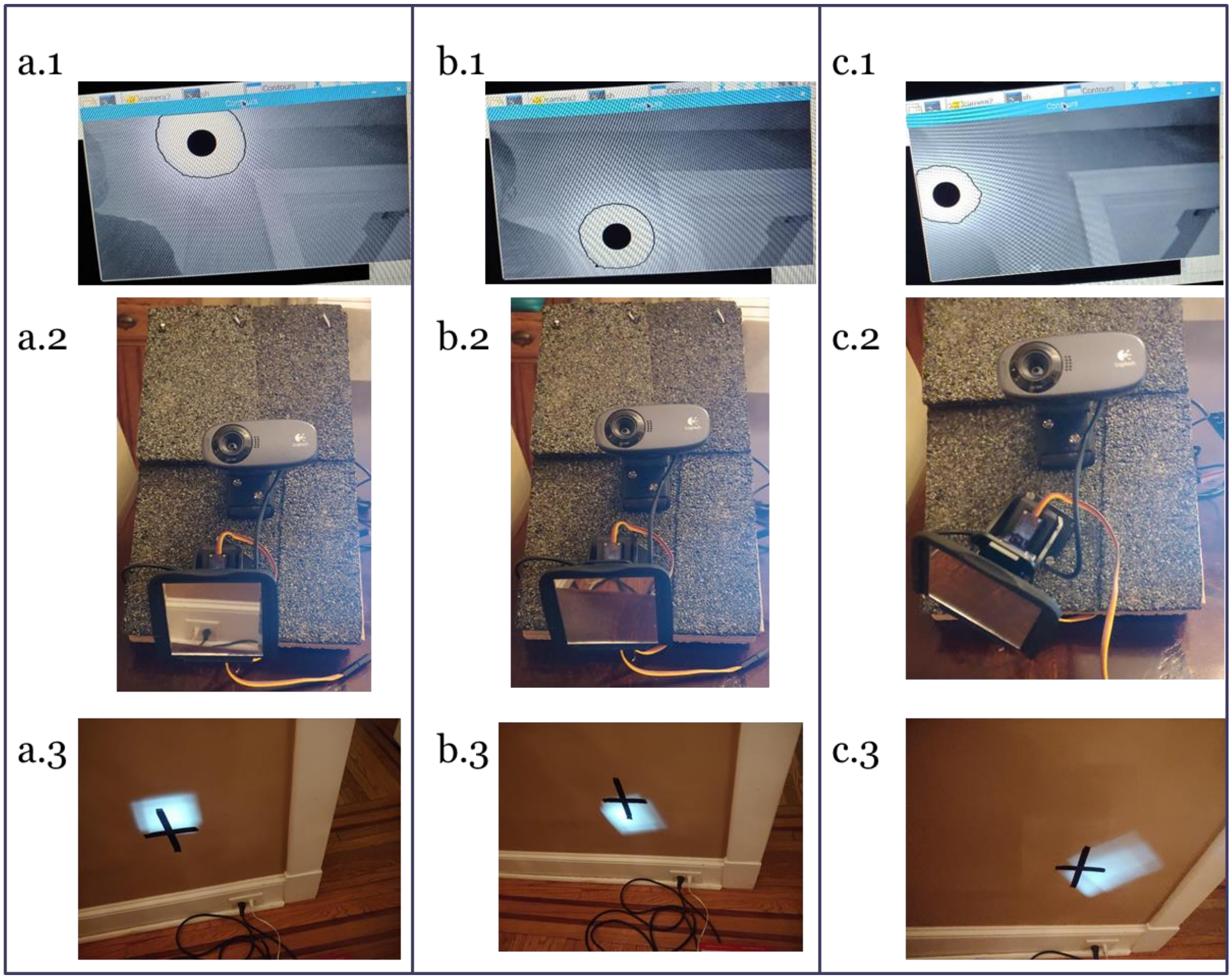

Fig. 4 Prototype experimental test results, simulating a.) a sun image at a high elevation, b.) a sun image at low elevation, and c.) a sun image at medium elevation and shifted to the left of the view camera and mirror (from the perspective of facing the camera/mirror).

As depicted in the results of Figure 4, the aiming point of the reflection relative to the position of the camera was accurate (i.e. it consistently hit portions of the target and where it was off-target, the error was consistent), but not precise (ideally the reflected light would always land on the center of the cross).

The less than ideal precision was attributed to error stack-up. Specifically, any imprecision in the camera field of view measurements and any misalignment in the camera and mirror mounts with respect to each other as well as the input values used in the software, and any imprecision with respect to the pre-determined target position relative to the camera/mirror system, had a multiplicative effect on the reflection pointing error. In light of this error, the results were found to be satisfactory. Additionally, adding more mirrors to form an array would broaden the swath of covered area, making the error even less significant to achieving the desired effect.

Conclusion and Future Work

Although these components currently function, we have not mounted the system on an actual roof outdoors to test the efficiency of using this method to grow grass in shady areas.

Potential difficulties in our system are very similar to those for any solar tracker, including clouds, fog, temperature fluctuations, and wind [1]. Since our solar tracker uses the brightest object in the field of view and operates under the assumption that this object is the sun, we need to control for the albedo of objects that are highly reflective, such as passing car windshields or windows from a neighbor’s house. Because we know that the sun moves slowly across our field of view, adding a buffer (e.g. 5-10 minutes) to the system to recalculate the brightest object prior to adjusting the mirror would resolve some of these inconsistencies, since in most instances these surfaces would stop reflecting light within that time frame and the sun would again become the brightest object. Weather conditions could also affect the brightness of the sun, so that 200 may be too high a threshold to capture the sun as an object on an overcast or cloudy day. Future work on this project would involve calculating the best threshold to use regardless of weather conditions rather than using a fixed set-point, although it can be argued that if the sun is not bright enough to detect using a reasonable threshold, there would be limited benefit to redirecting the sun’s rays at that time.

Once the best threshold issue has been solved, this device has multiple potential applications outside of the proposed scope. Solar tracking has the potential to be used not only for the benefit of yard aesthetics, but also to direct and harness solar energy for other purposes.

Residential food production could be increased by redirecting sunlight to shady spots of a residential yard or community property that was previously unable to sustain a backyard garden. On a bigger scale, solar tracking could be used to redirect sunlight towards urban farms/gardens where land patches are often shaded by large buildings throughout the day.

In the power generation industry, solar tracking can be used to tilt solar panels to follow the path of the sun throughout the day, increasing the number of direct rays captured and thereby increasing energy production of each solar panel. This could be utilized on a small scale by home owners looking to sell energy back to the power utility company, or by large scale power suppliers themselves. Solar tracking is also used in solar cooking, so that the hot surface does not need to be continually adjusted to focus the sun's rays. Solar cooking is especially important in impoverished nations, which lack more traditional fuel sources such as natural gas or wood. In areas without access to clean water this becomes even more important, as water must be boiled prior to consumption. In these ways solar tracking can be used to improve the health and safety of humans and the environment.

References

[1] W. Nsengiyumva, S.G. Chen, L. Hu, and X. Chen. "Recent advancements and challenges in Solar Tracking Systems (STS): A review," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 250-279, Jan. 2018.

[2] A. Rosebrock, "Raspbian Stretch: Install OpenCV 3 + Python on your Raspberry Pi," Sept. 2018, https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/