SE 101: Introduction to Methods of Software Engineering

University Of Waterloo

Software Engineering

Candela: Light Tracking and Path Retracing

Sherwin Aminpour, 20529526, saminpou

David Dong, 20518392, c9dong

David Fu, 20516684, d23fu

Jeremy Hidajat, 20504938, jhidajat

Tony Lu, 20512897, dslu

Jason Loo, 20516283, jmloo

Richard Wong, 20511561, rk2wong

November 28, 2013

SE 101 Team 3

University of Waterloo

Waterloo, ON

N2L 3G1

November 28, 2013

Krzysztof Czarnecki, Instructor
Software Engineering
University of Waterloo
Waterloo, ON
N2L 3G1

Dear Sir,

The enclosed report, entitled “Candela: Light Tracking and Path Retracing”, is related to the final project of our SE101 course. This project required us to program a Scribbler 2.0 robot to perform a specific task, and to document the process of generating the final product.

The primary functionality of our robot is to detect and follow a light source, then retrace its path back to its original position once it loses vision of the light source.

This project integrated knowledge of sigmoid function variants, sensor calibration techniques, and input processing in order to achieve a product that would perform effectively in various environments.

We hereby confirm that we have received no help, other than what is mentioned above, in writing this report. We also confirm this report has not been previously submitted for academic credit at this or any other academic institution.

Sincerely,

SE 101 Team

Table of Contents

Executive Summary

The global effort to seek sustainable sources of energy has thus far made many significant advances and has brought forth more opportunities for innovation. One blooming prospect is in the development of solar power collection and wireless power transmission from satellites orbiting the Earth [1]. Already, NASA has developed proof of concept of the viability of wireless power transfer [2]. Combined with operational solar power collectors, space solar power may become a viable alternative or even a replacement for conventional means of power production.

One subproblem in producing an autonomous system for solar power collection is that of maximizing light exposure, which consequently maximizes total power output [3]. Thus, it would be optimal for the system to locate and point towards a location that provides maximum light exposure. Our group’s approach to this problem involves developing a Scribbler-based system that utilizes variable light sensor readings to differentiate between regions of high and low light intensity, then calling forth an appropriate motor response from the Scribbler to move and point towards the light.

Additionally, light-tracking behaviour can be implemented in applications such as in the exploration of dark environments. The programmed behaviour would thus also include the ability to follow a moving light source, as in an explorer carrying a luminescent marker. In such a scenario, it may be desirable for the robot to retrace its path to assist lost individuals return to their origin within a reasonable degree of accuracy.

This report documents the group’s approach to the aforementioned problem and alternatives made to deal with system limitations, and provides an analysis on the solution’s degree of success based on design specifications.

List of Figures

List of Tables

1 Problem Specifications

1.1 Functional Requirements

1.1.1 Determine the Brightest Light Source

Initially, the robot must be able to determine the brightest light source in its vicinity. The robot should accomplish this light assessment by receiving brightness readings from its surroundings without moving too far from its original location. Upon detecting the brightest light, it is necessary for the robot to alter its orientation until it is facing the light source. If the robot fails to find the brightest light source, the light commands will be unable to manipulate the robot's movements.

1.1.2 Follow Light Source

The robot must follow the given light source. The robot is expected to regularly scan the brightness in its field of vision. Based on the intensity and position of the light readings, the robot should either continue along its path of motion or adjust its alignment so that it faces the light source directly. However, it is required that the robot maintains its constant speed while adjusting its orientation. Hence, the modification of the robot's direction will be gradual. During this stage, the robot must track all of its motion data and store it into a data structure for later use. If the robot fails to follow the light source, it could deviate from its proposed path.

1.1.3. Recognize Absence of Light Source

Without a terminating condition, the robot would indefinitely follow what it perceives to be the brightest light source. A functional requirement is for the robot to cease following the light source once the light source is removed. To achieve this, the robot must store a brightness reading to compare to the brightness reading of the light source. Upon removal of the light source, the robot should recognize that the subsequent brightness readings match the origin brightness reading and it should proceed to the final phase of it’s performance.

1.1.4. Retrace Path

In the final stage of the program, the robot is required to remember the path that it had taken and retrace its steps. The robot must interpret the motor data that it had previously saved and apply it in reverse order until there are no more available commands. Upon completion, the robot should be at its starting position.

1.2 Non-Functional Requirements

1.2.1 Response Time

There should be little to no delay between the interpretation of the light source brightness and the input to the motors. The robot should recognize that the light source has been removed within a reasonable time limit (for instance, within three seconds).

1.2.2 Testability

All components of the robot must be able to be tested individually in order to determine their capabilities and accuracy. Each phase of the demonstration should be able to be tested independently of one another. The results of the tests must be consistent, thorough complete.

1.2.3 Modifiability

The code should be written in a way such that it is easily modifiable. One example of this is using constants instead of number values throughout the program to make slight modifications after testing. It should also be easy to add new features to the project.

2 Design Constraints

2.1 Sensors

The main constraint for this project was the accuracy of the light sensors. Prior to the coding, a test was carried out to determine the reliability of the light sensors. The test was done by positioning the scribbler robot inside a dark room with nothing but a single fixed light source shining at it. The robot then turned 360 degrees counter-clockwise and outputted the difference between the rightmost brightness sensor reading and the leftmost brightness sensor reading. The graph below was obtained:

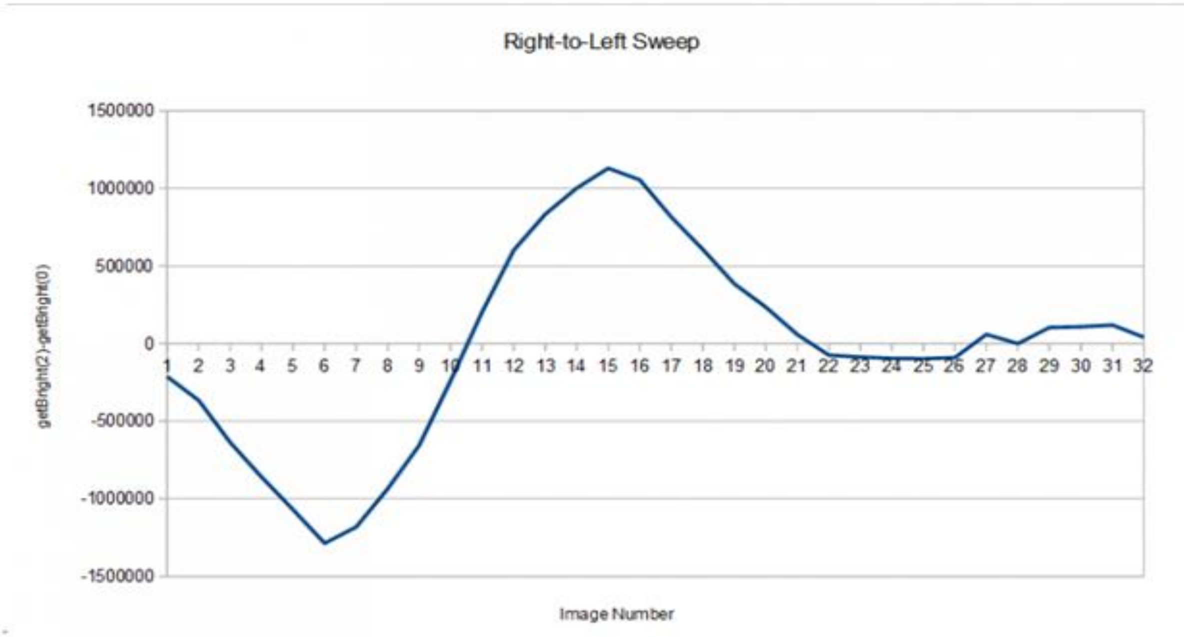

Figure 2-0-1. *Differences in Light Sensor Readings in Right and Left Fields of View*

As illustrated by the graph, a function similar to the negative sine function was obtained. Two obvious problems can be observed from the graph. One of which is that the graph is composed of line segments as opposed to smooth curves. This shows that the scribbler robot is incapable of rotating and detecting light at the same time. As a result, the robot must first rotate before it can take readings with its light sensors with each iteration. Small turning angles would maximize accuracy, while large turning angles maximize time efficiency. Thus, several tests were performed to optimize accuracy and efficiency, culminating in a method which would allow the robot to dynamically determine the optimal turning angle.

The second problem is evident in the right half of the graph. The group expected to see a near-horizontal line overlaying the x-axis when the robot faced away from the light source, since theoretically the robot should not have detected light in its dark surroundings. However, as seen from the graph, several local maximum and local minimum points were returned instead. This could have been caused by the interfering light from the environment, the internal error of the sensors, or a combination of both. The final code was therefore tailored to accommodate for this signal deviation.

2.2 Movement

Another significant constraint is the inaccuracy of the robot's movement. Several tests were conducted to determine its precision. The result showed that the robot's movement was affected by both the surface friction and battery level. For example, on a cement floor, it required approximately 3.15 seconds to turn 360 degrees, but on a carpet, it took approximately 3.5 seconds to turn 360 degrees. There was also slight inconsistency with the speed of the robot moving forward and backward even when the input speeds were set the same. Movement accuracy is crucial to the success of the retracing stage of the project because the robot must be able to return to its initial position with reasonable accuracy.

2.3 Environment

Dynamic and unpredictable elements of the operating environment may induce undesired behaviour. For instance, dynamic terrain and lighting interfere with retracing movement and undermine signal calibration. Additionally, for the purpose of complying with the problem description, the group does not need to consider significant changes in environmental lighting. Lighting conditions in terrestrial orbit, facing the sun can be assumed to remain relatively constant.

2.4 Others

Additional constraints include delayed response time from the robot and LED lights from the robot producing signal noise. However, these constraints are trivial and rarely affect the performance of the robot.

3 Main Design Challenge

The main design challenge for this project is to program the Scribbler robot to accurately detect the position of the light source and follow it. Due to the limitation of the light sensors built into the robot as mentioned in the design constraints, it is very difficult for the robot to maintain a consistently accurate detection of the light source. After extensive testing, four solutions were proposed.

3.1 Look and Turn Method

The Look and Turn Method is a very simple algorithm that reads the output from the left and right light sensors and compares them. If the right light sensor value is greater than the left light sensor value, then the robot would turn right for a set period of time. If the left light sensor value is greater than the right light sensor value, then the robot would instead turn left for a set period of time.

3.2 Sigmoid Function Conversion Method

The Sigmoid Function Conversion Method utilizes the "S" shape of the sigmoid function to convert the difference between the left and right light sensor readings into valid motor inputs. The equation and graph of the sigmoid function are outlined in **Figure 3-1**.

$$S(t) = \frac{1}{1 + e^{-t}}$$

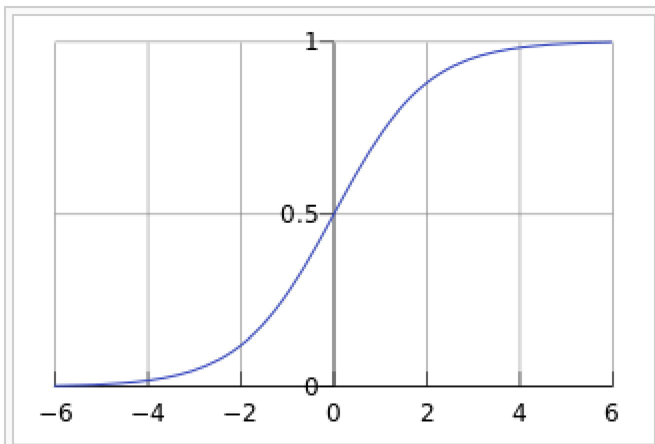

Figure 3-1. *Logistic Function Equation and Graphical Representation*

The output of the sigmoid function always lies in the open interval (0, 1). If a positive value is supplied to the function, then a value in (0.5, 1) will be outputted. Likewise, if a negative value is supplied, the a value in (0, 0.5) will be outputted. Given this pattern, the left motor value could be outputted by inputting the positive difference between the left and right light sensor values, and the right motor value could be outputted by inputting the negative difference. This would in theory allow the robot to dynamically change its turning angle based on the intensity of the light.

3.3 Sigmoid Function Conversion Method with Positive Signal Bias

This method uses the same logistic function for signal processing as the Sigmoid Function Conversion Method. However, the input to the motors is governed not only by the differences between the left and right sensor readings, but also the magnitude of the middle sensor reading. Thus, the robot would respond appropriately if it detects a light source in front of it, but the difference in its left and right sensor readings is close to zero.

3.4 Sigmoid Function Conversion Method with Positive Signal Bias and Redetection

This method combines the positive signal bias algorithm from the method described in Section 3.3 but includes an additional operation stage such that when the robot loses vision of the light source, it performs a 360-degree sweep of its surroundings in an attempt to recover vision of the source. This allows for a much more accurate light detection at the cost of performance speed.

4 Success Criteria

To evaluate the four solutions proposed, three criteria were specified as shown in **Table 4-1**:

Criterion	**Criterion Description**	**Weight**
Accuracy	The robot must be able to face the light source with an acceptable error of +- 5 degrees	30%
Time	The robot is able to complete its task in a swift manner	20%

Utility	The functionality of the robot can be applied to real world needs	50%

Table 4-1. *Criteria for Success and Criterion Weightings*

These three criteria were used to evaluate the degree of success with which the solution carries out the task outlined by the problem description. After extensive testing, the following results in **Table 4-2** were obtained:

Method	**Accuracy Score (/ 30)**	**Time Score (/ 20)**	**Utility Score (/ 50)**	**Total Score (/ 100)**
Look and Turn	10	12	20	42
Sigmoid Function Conversion (SFC)	20	17	42	79
SFC with Positive Signal Bias	25	16	42	83
SFC with Positive Signal Bias and Redetection	28	10	35	73

Table 4-2. *Computational Decision-Making Matrix*

The Look and Turn Method scored the worst out of the three in terms of accuracy. The Sigmoid Function Conversion Method scored the best in terms of time, and had a moderate accuracy. The Sigmoid Function Conversion Method with Positive Signal Bia and Redetection scored the best in terms of accuracy, but had the worst time.
In terms of utility, the SFC Method and SFC with Positive Signal Bias Method were determined to be the most useful because they accomplished the task of following a fixed or moving light source with little to no wasted movement. The SFC with Positive Signal

Bias and Redetection Method offered a much more accurate result, but a situation that requires the robot to relocate the light source is rare, and in extreme conditions, such action could even cause the robot to deviate from its original path. The Look and Turn Method failed to follow the light source on multiple occasions, therefore it is the least useful out of the four methods.

Based on the weight values, the Sigmoid Function Conversion Method with Positive Signal Bias was chosen to solve the main design challenge.

5 Design Analysis

The goal of the project was to design an autonomous system that would track and follow the brightest light source in its surroundings then return to its original position upon losing vision of the source. The group evaluated the robot's ability to perform these tasks using criteria involving accuracy and time efficiency. Considering the utility of our design, where a robot must operate in a high contrast environment to find a source of light, the group chose not to consider bright operating environments, which contribute a significant amount of signal noise. Therefore, the robot's performance in bright areas can be excluded from the success criteria, as its performance is considered unrelated to the robot's ability to operate under conditions as specified by the problem description.

5.1 Accuracy

The performance accuracy of the robot can be determined by two factors: the accuracy of its sensor readings, and the accuracy of its motor movements. Through trial and analysis of data, we determined that the robot is able to distinguish between brightness variations through its left and right sensors with moderate variation. Throughout all detection methods, the primitivity of the light detectors impeded the robot's ability to correctly determine the brightest light source in its surroundings. Some of the ambient light was reduced by covering the lights that were emitted from the robot’s LEDs. However, the built-in light detection functions seemed to favour the physical size of light sources (e.g. walls) rather than their light intensity.

Under dark conditions, the robot tracks light almost perfectly. The robot can correctly determine ambient brightness through its initial baseline calculations so that after the light source is disabled, it will execute its return procedure with minimal delay. The robot’s field of vision is narrow (approximately 20 degrees), however. This somewhat restricts the robot’s vision accuracy when the light source moves too quickly or too abruptly.

The movement accuracy is dependent on the motor power and opposing frictional force. Since fully charged batteries yield strong motor force, the robot must be kept at full battery life to ensure proper functioning. Additionally, since frictional force varies from surface to surface, the robot must assess its environment before execution to determine how much motor force is required to move and turn for any given distance. To accommodate this, the robot performs diagnostic rotations before every execution to determine with what force each rotation must use to rotate a fixed angle. This greatly reduces the movement error and thus increases the robot’s overall accuracy.

5.2 Time Efficiency

Using the Sigmoid Function Conversion Method, the robot did not have to make unnecessary stops after every scan of its environment. Instead, the motors would continuously function in real-time and adjust to the output of the brightness detectors after they were run through the sigmoid function.

The method used by the robot to communicate with the computer using Bluetooth went through several design iterations. Before, the robot would return left and right brightness sensor readings separately throughout the execution loop. The wireless communication contributed a large amount of time to the delay between each action the robot would make, such as turning or adjusting its motor speeds. The current method of communication reduces the number of times data needs to be transmitted between the robot and the computer by combining the output of the brightness functions into a single array that is sent wirelessly.

5.3 Conclusion of Results

The main flaw suffered by the robot's brightness detection accuracy is its narrow field of vision. However, this is mainly impeded by the physical restrictions of the robot, not through any design flaw. Recalling the utility criteria, this is not a large problem in real situations since light sources are typically stationary or very slow (for instance, the sun passing through the sky). Given these circumstances, the robot is still able to turn correctly according to the relative position of the light source, and at rates that are proportional to the distance the light source deviates from the center of its field of vision while compensating for terrain impedance. Therefore the robot proves to be sufficiently accurate on all other accounts.

The robot was able to perform its tasks in real time, without any pauses or abrupt, staggered movement. The robot spends a moderately large portion of its execution time performing diagnostic tests, though they greatly improve the accuracy of the results and are ultimately critical to the robot's performance and deemed necessary.

The limitations of the robot are derived mainly from the physical incapabilities of the robot. For instance, faster motor speeds would allow sharper turning and therefore greater competence in detecting and following light sources. A wider field of vision would produce similar results. Additionally, many of the time efficiency problems would be eliminated in real-world applications. For instance, the computer does not need to be remote from the physical robot, thus wireless communications are not necessary and performance times can be severely reduced.

References

[1] M. John, "SPS-ALPHA: The First Practical Solar Power Satellite via Arbitrarily Large PHased Array", (NASA), [online] 2013, http://www.nasa.gov/directorates/spacetech/niac/mankins_sps_alpha.html (Accessed: 25 November 2013).

[2] Space Solar Power, *National Space Society*, [online] 2013, http://www.nss.org/settlement/ssp/ (Accessed: 25 November 2013).

[3] K. Jim, "Maximizing Solar Panel Efficiency and Output Power", (ElectroSchematics), [online] 2013, http://www.electroschematics.com/8280/maximizing-solar-panel-efficiency-and-output-power/ (Accessed: 25 November 2013).

[4] Sigmoid function, *Princeton University*, [online] 2013, http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Sigmoid_function.html (Accessed: 25 November 2013).