Lab 3 – Bug Algorithms

Austin Holmes – U#31858364

Control of Mobile Robots – Section 001F23

INTRODUCTION

This lab, the third in the Control of Mobile Robots series, introduces yet another modality of information consumption, the color (traditional) camera, as well as combining the foundations built in previous labs to produce a serviceable algorithm which completes a task. Specifically, the wall following behaviors developed previously are combined with target-acquisition capabilities provided by the onboard camera to produce a simple "bug algorithm" which seeks out and approaches a recognizable target object.

TASK 1 – TARGET ACQUISITION

The first aspect of the bug algorithm, and the initial task of this lab, is to recognize a target object (by color) within the visual field. The target should also be sought out when it cannot be presently seen. This is achieved by using an OpenGL blob detection algorithm built into FAIRIS-Lite. The target acquisition process also incorporates an analysis which determines the angle of the object relative to the robot as well as the estimated distance based on visual and target information.

Once the target and related information is obtained, task 1 asserts that the robot should approach and stop within a specified distance of the object.

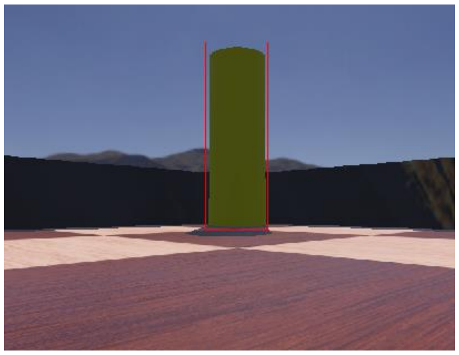

Figure 1 – Target Object as Seen by Camera

TASK 2 – BUG ZERO

The second and primary task in this lab is the implementation of the "bug zero" algorithm, which is described by the seeking of a target object, followed by an approach toward that object. When an approach is obstructed, the perceived obstacle will be circumnavigated until the target object is no longer obstructed. In the implementation described herein, this algorithm is broken into three decision patterns:

A. No Target Visible

B. Target Visible but Obstructed

C. Target Visible with Clear Path

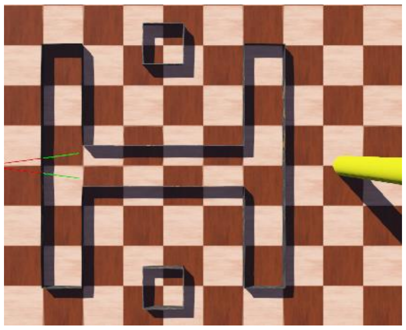

Figure 2 – Task 2 Environment 1

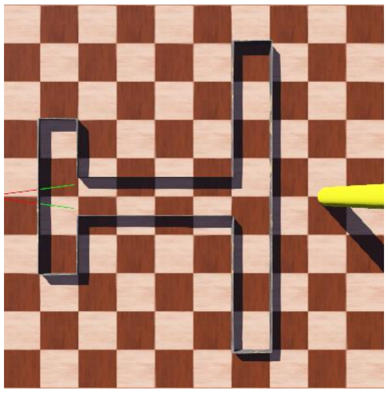

Figure 3 – Task 2 Environment 2

KEY VALUES

A. Target Size

As to enable visual analysis of target position based on distal scaling, the exact dimensions of the target are provided to the robot.

B. Camera Parameters

The experimentally-obtained horizontal field of view (FOV) and resolution are given as parameters to the robot to enable analysis of angular information produced by the camera.

C. Target Angle Width

Essentially another precision parameter, this represents the tolerance for target approach rotation. As long as the robot is facing the target within this margin of error, no rotation will be performed.

KEY FUNCTIONS

A. Find Target

Find target performs the bulk of new information gathering and analysis for this lab. Provided the camera interface, this produces an array of information including the distance from the robot to the target as well as its angle from forward relative to the robot.

The relative angle of the object is given by the following, where θ is the angle from forward, p is the center point of the target on the image, r is the horizontal resolution of the camera, and f is the FOV of the camera:

$$\theta = \frac{p}{r} - \frac{f}{2}$$

The distance from the robot to the object is given by the following, where d is the distance, s is the horizontal size of the object on the image, r is the horizontal resolution of the camera, and f is the horizontal FOV of the camera:

$$d = \left| \frac{\frac{s}{2}}{\tan\left(\frac{1}{2} f \frac{s}{r}\right)} \right|$$

TARGET APPROACH

The target approach goal is achieved by utilizing the "find target" function to obtain the angle and distance of the target, and then rotate until the target is forward of the robot and move the specified distance such that the robot stops within the specified stopping distance parameter. If the target is not visible (the function returns an empty array), the robot rotates by one FOV and then performs another target acquisition cycle. This continues until the target is reached.

BUG ALGORITHM

As described above, the bug algorithm uses the target acquisition function and target approach ability to perform the target seeking task according to these three conditions:

A. No Target Visible

In this case, the selected (right/left) wall will be followed. If the nearest wall is forward, the wall will be approached and a 90-degree turn will be executed to enable wall following.

B. Target Visible but Obstructed

In this case, wall following will also be executed with the same forward wall condition as the former case.

C. Target Visible with Clear Path

In this case, the approach methodology from task 1 will be applied.

NAVIGATION SCRIPT

As this lab does not require a specific path to be followed, navigation is handled by the bug algorithm itself. All paths taken are determined by the decisions made by the algorithm given the observable information of the environment.

CHALLENGES

The most challenging aspects of this lab were issues both new and old – wall following and photograph analysis. As was evident in lab 2, PID wall following is subject to wild variability depending on environmental and behavioral parameters, leading to a lack of reliability, especially given a clearly poor implementation which appears difficult to improve. Camera analysis has also introduced some issues, as when the target occupies the entire visual field, blob analysis becomes difficult as the library used does not identify the edges of the field of view as object edges, meaning an image consisting of entirely the target color is interpreted as essentially empty.

CONCLUSION

As the previous labs have laid the essential foundations, implementing simple behavioral algorithms is not inherently difficult, but does reveal every weakness, hidden and apparent, of its components. Wall following continues to prove difficult and in need of great amounts of optimization.

NOTES

The performance of the robot is far from acceptable and will require complete overhaul in several areas to support future lab tasks. For now, however, time runs short.

VIDEOS

Included in submission archive.