6.111 Team Proposal

Emmanuel HAVUGIMANA, Shuto OGIHARA

October 2019

1 Overview

Our project is about making a robot car that could chase an object using a loaded camera, and make it do some specific tasks using the chasing algorithm. To make it simple, the object we will be using would be a circle/sphere, and will be identified and located in a scene using morphological image processing techniques and size detection. Chase-bot will move to keep the object appear in the same location and the same size to keep the distance based on a control algorithm. We would also make it to get closer to the object and eventually collide. The basic task that we would implement would be to make it follow the a ball, which was determined by the initial scene by itself, attached to a string. We will also display what the robot sees for debugging and demo purposes. Ideally, the robot would be able to chase the object with no delay, which will enable it do some other interesting tasks. Some of the tasks we will implement would be, catching a rolling ball and bringing it back, or act as a goal keeper by replacing the camera to the side. The hardware would require some additional features accordingly.

2 Block Diagram

Camera takes an image and image is sent to track. The **Initialize** module would provide which object to track and the goal information to use. Tracker module would find a match to the object we are tracking, and if a match is found, it outputs the location and size of the match. The **Tracker** module will be implemented by binarizing the original image and the object to track on a color channel of interest. Object to tracker is matched to find where there is a maximum match and use that as the location of the object if object size in an image is smaller than object desired size. the maximum match will be small and tracker will output the difference of the desired match to value of the match. The controller module will use position differential information and size to move relatively as to minimize the differential. It would decide to either keep the distance or get closer to the object depending on the input. Our System is closed as the vehicle move, the image in the scene will change and our tracker

module will register a new position and size and controller module will get the updated values until the difference is small enough for tracking purposes.

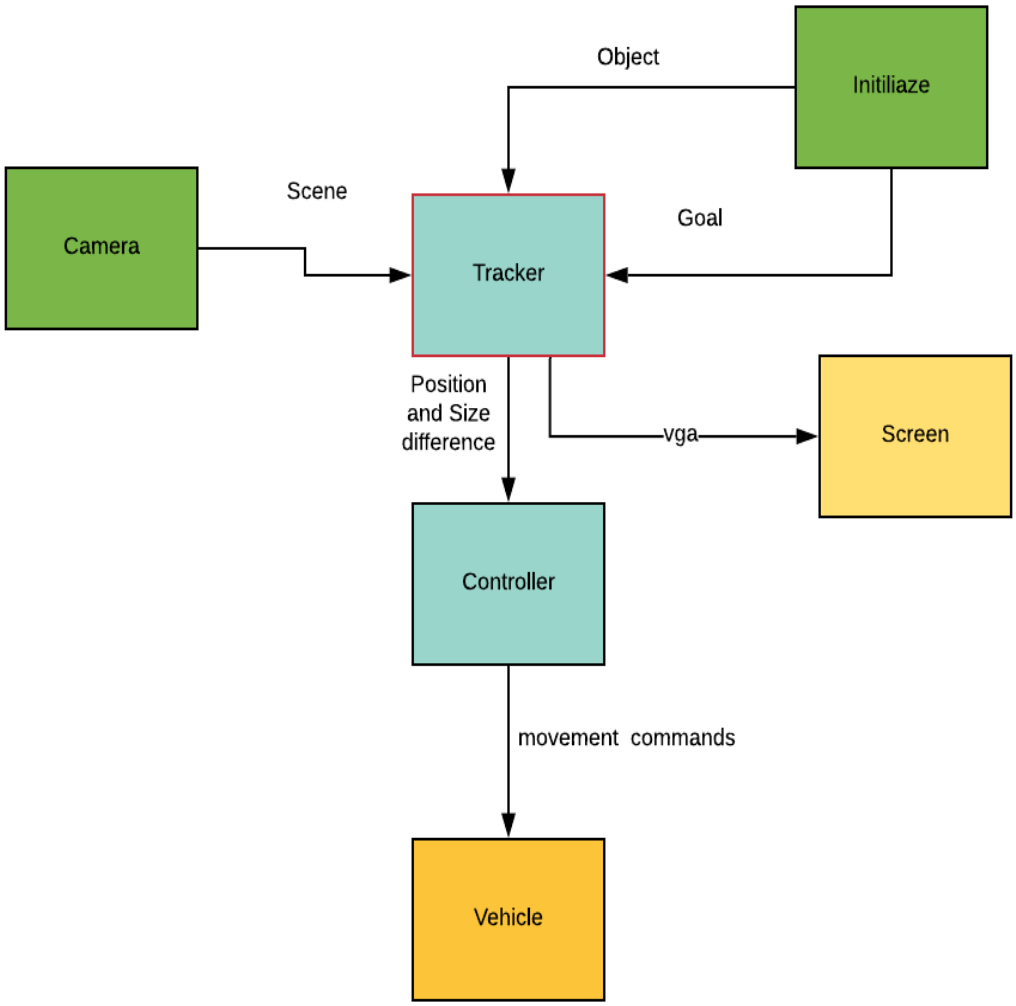

Figure 1: Block Diagram

3 Modules

3.1 Camera module

input : Image
output : Digital Image representation (640x480) at 30fps

3.2 initialize module

Output 1 : target parameters color, and size(ball target)
Output 2 : Goal size and horizontal position in image

3.3 tracker module

input 1: image from camera
input 2: target parameters
input 3: goals
output 1: signed position differences between goal and the localized target

3.4 screen module

input 1: image
input 2: position and size of target
output: display image and show tracked object

3.5 controller module

May be external or internal to the FPGA and translate position differentials and size into movement commands to the motors. The outputs would be the speed specified as a pwm signal, and direction for each motor.

4 Hardware Setup

We would purchase a DIY Robot smart car kit as the chase-bot, and load it with the camera. Our plan is to put the FPGA on the car as well to take advantage of the low latency of FPGA. The car would have two motors with speed encoders, and their direction and speed could be controlled. The flow of controlling the motor would be as shown in figure 2. The L289N would be used to drive the motor, and it will have three inputs; pwm, In A, In B, for each motor. The pwm would control the speed of the motor by altering the duty cycle, and In A and In B would control the direction of the motor to spin. For instance, making the In A high and In B low, and the pwm as a duty cycle of 0.5 would make the motor move forward in half of its maximum speed. In order to accomplish the fetching task, we would have an external hardware to install on the car which would be added after the basic functions of the project was accomplished. The design of our final hardware would look like figure 3 and figure 4.

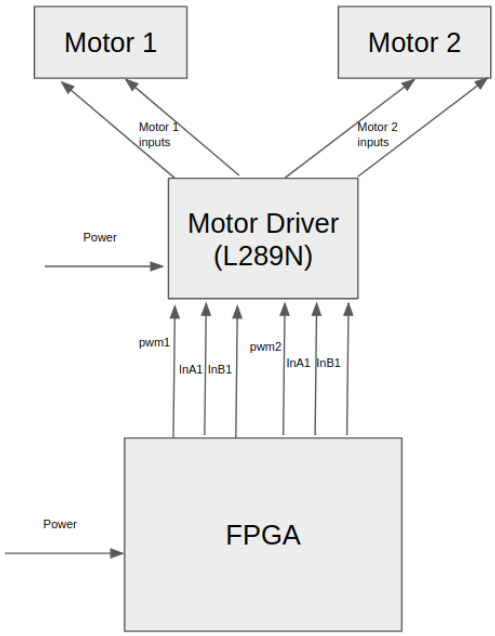

Figure 2: Controlling the motor

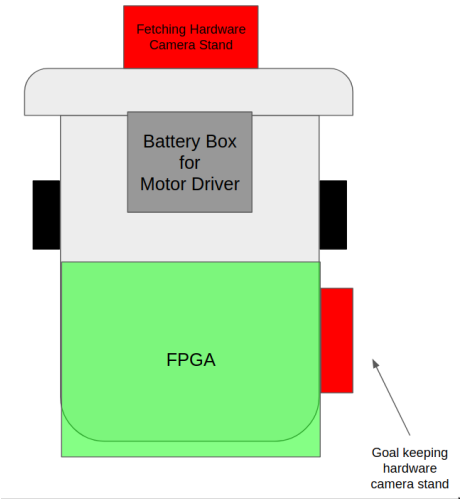

Figure 3: Controlling the motor

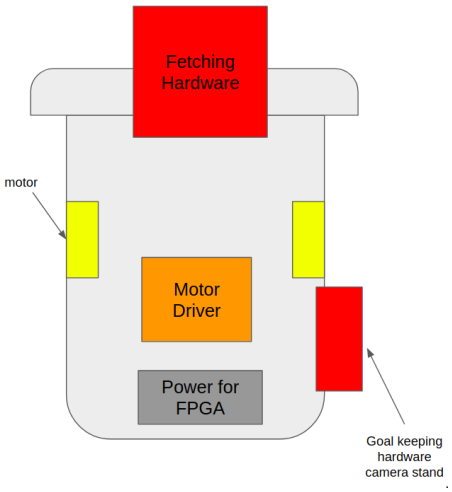

Figure 4: Controlling the motor

5 Goals

5.1 Baseline

- Following a ball based on its camera input
- Initializing the screen to get the object information

5.2 Expected

- Calculate distance based on camera input and object physical size
- Goal Keeping Task

5.3 Stretched

- Fetching Task
- Self-estimation based on the speed encoder information
- Drawing its path on screen
- Prediction of the ball trajectory

6 Contributions and Schedules

The planned timeline of the project is shown below accounting for the deadlines based on the website information. Whichever having the name on the task would be responsible for completing the task, although some work could be done together.

- **10/28 Project Proposal**
- 10/28 - 11/04 Block Diagram[**Shuto**] & Camera test[**Emmanuel**]
- 11/02 - 11/08 Hardware[**Shuto**] & Tracking module[**Emmanuel**]
- **11/05-11/07 Project Design Proposal**
- 11/09 - 11/15 Complete Tracking[**Emmanuel**] & Control, module[**Shuto**]
- 11/16 - 11/22 initial object identification[**Shuto**] & Experiments[**Emmanuel**]
- 11/23 - 11/27 Goal Keeping Modules[**Emmanuel**] & Hardware[**Shuto**]
- 11/28 - 12/8 Fetching Modules [**Shuto**] & Report, Demo Preparation
- **12/9 Final Project Checkoff**
- **12/10 Demo, Video**