

Walligator

Donald Burnette

University of Florida
Department of Electrical and Computer Engineering
EEL 5666 - Intelligent Machines Design Laboratory

(April 20th, 2007)

Table of Contents

Abstract	3
Executive Summary	3
Introduction.....	4
Mobile Platform.....	4
Actuation.....	5
IR Sensors	5
Sonar Sensors.....	6
CMUCam2.....	7
Behaviors	8
Conclusion	8

Abstract

The robot is designed to speed around a track using wall following methods as a guide. The robot will use various custom techniques for finding and tracking the midline between two surrounding walls. In addition, the robot will navigate turns quickly and smoothly, without having to slow down and without veering off course. The robot will perform these tasks using on board IR and sonar range finders. One of the longer straight sections of the course, the lane will diverge into two lanes separated by a wall in the middle. Each lane will be associated with a given color, and the robot will select its path based on an LED sitting on the center wall. The color of the LED will be detected by an onboard CMOS camera in real time, as the robot approaches the fork. Typically, small, microcontroller robots tend to respond slowly when faced with real time vision processing due to the immense amount of data that needs to be processed by a relatively slow processor. Using the new CMUCAM2 and innovative detection techniques, it is the goal of the project to be able to detect and process the color of the LED quickly as the robot approaches, without having to slow down or hesitate. The robot should be able to navigate the entire course, including the two way fork, cleanly without having to slow down and without showing any signs of heavy computation.

Executive Summary

Walligator was constructed for IMDL during spring 2007. The original intent was two-fold. First, I wanted the robot to be able to navigate a rectangular track seamlessly using IR sensors to wall follow. Second, I wanted the robot to be able to detect a color using an on board camera whiling moving at full speed, and then make a decision based on that color.

The vision was more or less a success, and the robot is able to detect a color sign and decide to go left or right based on the color that it sees. The first task however, proved much more difficult. The IR sensors are not very reliable and they interfere when placed in close proximity. Future efforts to wall follow in this manner should turn toward side looking sonar sensors instead, as I had better results using sonar. I was not able to run Walligator at the highest speed setting, but it goes faster than most other robots trying to do a similar task, and certainly faster than other robots doing vision procession in real-time.

Introduction

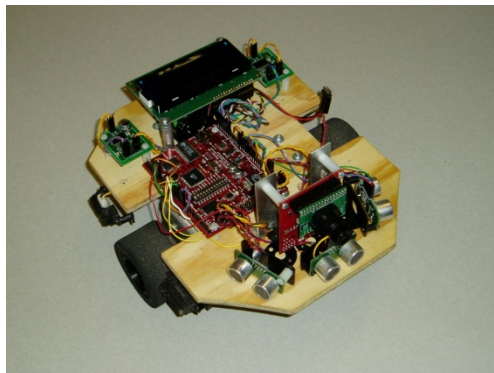
After witnessing many robot demos involving real time vision, I was not satisfied with the current state of vision processing at the embedded microcontroller level. I am convinced that techniques can be developed to handle and filter large amounts of vision data quickly and efficiently using only an onboard microcontroller. This rather simple robot will serve as a proof of concept that it is possible to process, filter, manipulate and respond to large quantities of data in real time.

Mobile Platform

The robot consists of various components: the frame, the drive system, the sensors, and the LCD screen. The frame was first constructed out of $\frac{1}{4}$ " condensed pine wood, which is strong and rigid, as a test platform. The frame, once finalized, was to be then cut out of $\frac{1}{4}$ " clear polycarbonate. Time restraints became the determining factor, and the polycarbonate was never cut out.

Below the frame is the wheel mount, which is just a strip of aluminum that was cut out to size and bent to form a U-shape. The motors are directly mounted onto this aluminum chassis.

The four IR sensors are mounted around the perimeter of the frame; three sonar sensors are at the front, along with a CMUCAM2 mounted in the front atop the frame. The main board, a Mavric -IB, sits in the middle of the frame between the LCD and the CMUCAM2.



Actuation

The robot will use two dc gear head motors to controller the differential drive system. The motors chosen are model GHM-04 from HSIANG NENG. The rated voltage is 7.2V with a 50:1 gear reduction ration maxing out at 175 rpm. This model features a USDigital encoder shaft for use with the USDigital QME-01which features 480 quadrature counts per revolution. The motor/encoder pair is shown in the figure below.



Attached to the shaft are two high quality neoprene Blue Dot Sumo Tires 2.5" in diameter.



IR Sensors

IR sensors consist of an IR emitter, which emits a burst of IR photons, and an IR detector, which detects the concentration of photons returning to the device. As the concentration of return photons increases, the more current flows through the device, resulting in a higher output voltage. Two short range IR sensors (1.5"-30") are located on the sides of the robot, used for

precision wall detection. The wall centering algorithm will rely primarily on these two sensors in order for the robot to determine if it's centered between the walls. One long range, forward looking IR sensor (4"-80") will be used to detect upcoming walls and/or used for basic obstacle avoidance. The IR sensors used are the Sharp 2D120 and Sharp 2D12.



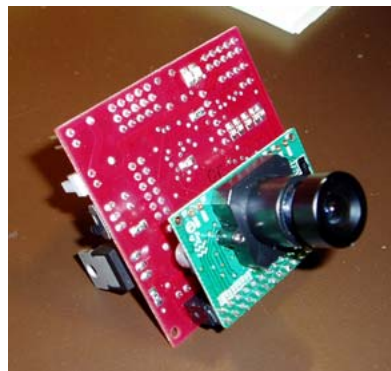
Sonar Sensors

Sonar plays the same role as the IR sensor, except it has a slightly different purpose. I have found sonar to be more reliable at far range, so I will be using diagonal looking sonar sensors to detect corners in the wall, in order to signal a turn is coming. Sonar works by first emitting a 40 Khz sound wave, which bounces back and is received by a microphone. A pulse is used to initiate the sound wave, and the sensor echoes a pulse when the sound wave has returned. It is up to the user to calculate the time between pulses. The longer the pulse, the farther the distance to closest object. The SRF05 sonar model was used on this project.

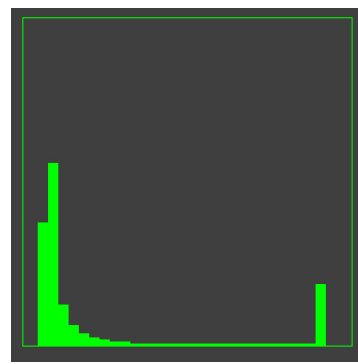
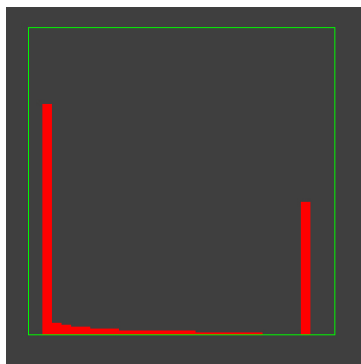


CMUCam2

I decided that the image processing will be done using a CMUCAM2 with an OV7620 color VGA camera. This camera was chosen for its quick and easy interface, along with its superior camera performance. The camera is capable of full color VGA resolution at 50 fps, allowing high image quality capturing even while the robot is in motion at full speed. Ultimately, I determined that the only relevant measurement from the CMUCam for determining color is the



histogram data, which is shown below. The histogram shows the number of pixels with a given concentration of the particular color, with the very left bar being zero color, and the very right bar being 100% of that color. I noticed quickly that the CMUCam seems to have a deficiency when it comes to detecting blue, as it would not correctly detect blue even when a plain blue image was put in front of the camera. I thus decided that red and green would be the two colors I used for my real-time decision making. I have tested the performance of the CMUCam in



moderate lighting conditions with two LEDs of the given color approximately 3 feet away from the camera, with the results shown. The difference is sufficient for a decision statistic when the robot is stationary, but when it's moving, the LEDs don't cut it. I ended up going with a color sign with a red side and a green side. When neither color is on, both graphs show very little color.

Behaviors

The robot will employ very basic behaviors, almost all primarily focused on avoiding collision with walls and other objects. The robot should be able to track a wall to either side, decide where the midpoint between those walls is, and quickly follow that midline parallel to the walls. The robot should employ smooth turns in EITHER direction, which adds an additional factor of difficulty. Traditionally corner algorithms are optimized for a single direction, and perform poorly or not at all when turning in the opposite direction.

Finally, the robot should be able to decide between two divergent paths given a color indicator detected using the onboard camera. The robot will not need to slow down or hesitate as it approaches the form, but instead will detect the path, and smoothly translate to the correct side and continue on. This was not possible to do at 100% speed, but ultimately the robot runs around 80%, which is still relatively fast.

Conclusion

In summary, Walligator successfully navigates walls along a roughly straight track, and can successfully use vision to detect the color a sign in real-time upon approach and then react based on that decision.

With that said, many improvements could still be made. If sonar sensors replaced the IR sensors as side looking sensors, more accurate wall following would be possible, and the robot speed could be increased.

The frame of the robot would look a lot better if it was made out of polycarbonate instead of wood, but this would just be show, and is not vital to functionality of the robot.

Finally, I'd like to say that almost every aspect of this project, from conception to completion, was much more difficult than I had expected. The sensors give noisy data, which you cannot always rely on, and you must prepare for. Sensors will break, so having backups is vital. I never anticipated this project taking up as much time as it did, and I am very fortunate that I finished in time.