Autonomous Driving System

Sensor Report

John Wernsing
March 17, 2005

EEL 5666
Intelligent Machine Design Laboratory
Dr. Arroyo & Dr. Schwartz

Table of Contents

Abstract

The ultimate goal of this project is to create an autonomous driving system for a car. Technologies, such as cruise control, already exist to aid drivers under monotonous conditions. Likewise, my system isn't meant to replace the driver, but merely make it easier and safer to drive.

The actual system builds on the traditional cruise control idea, but has the added benefits of automatically controlling the steering and speed of the car. It has the ability to find the lines on the road and to keep the car within the lines while moving. Additionally, it utilizes range finding capability to locate any cars or other objects in front of the car and to slow down and stop in response to those objects.

For this project, I am retrofitting a normal-sized remote control (RC) car with the sensors and circuitry necessary to implement the autonomous driving system. Sensors include a wireless video camera, a wireless link between the car and a nearby computer, an ultrasound range finder, and an infrared sensor located on the bottom of the car. As the car's circuitry is not powerful enough to perform the video processing on its own, a nearby computer provides the majority of the processing power for the system.

The system will be considered a success if the RC car can successfully navigate a road course completely autonomously.

Introduction

The autonomous driving system relies solely on its sensors in order to "drive" the car. Numerous algorithms process continuous streams of sensor data in order to instruct the car what it should be doing. This report provides a detailed summary of the sensors and some of the algorithms of the system.

High-Level Overview

The system exists as three distinct components continuously working together.

The first component of the system is the RC car itself. The RC car is normally sized (approximately 11" x 5") and contains all of the movable parts and majority of the sensors. The first sensor is an ultrasound range finder and is located on the front bumper of the car. It allows the car to determine the range of objects in front of it. A continuous stream of ultrasound sensor readings would also allow the car to determine the relative speed of the object. The second sensor is an infrared emitter/detector located beneath the car. Its function is to determine when the car is placed on the ground. The third sensor is a wireless transceiver located by the circuitry. The transceiver allows the three components of the system to quickly and reliably communicate with each other. This is essential as majority of the processing is performed in the other components. The last sensor of the car is a wireless color video camera located on the front of the car facing towards the road in front of it. It is the primary sensor and is responsible for finding and tracking the lines on the road. Other than sensors,

the car includes a servo controlling the angle of the two front wheels and a motor controlling the speed of the two back wheels.

The second component of the system is a wireless bridge that allows the computer to communicate with the car. As the wireless transceiver does not use a recognized computer protocol, the bridge must act as a liaison between the two components. It has a small microcontroller that translates and forwards information from the wireless transceiver to the serial transceiver. The bridge is connected to the computer through a serial port.

The last component of the system is a computer powerful enough to perform the video processing algorithms. The other two components do not have anywhere near enough processing power to perform them on their own. In fact, all of the “thinking” in the system is performed solely on this component. The other two components merely act as the executers and forwarders of the instructions.

Sensors

Ultrasonic Range Finder

The ultrasonic range finder allows the car to determine the range of objects in front of it. For this project, we use the “SRF04 – Ultra-sonic range finder”. It comprises of a small ultrasound transmitter / receiver pair that can be used to send an ultrasound “ping” and to measure the interval till the ping is echoed. As sound waves travel relatively slow, the microcontroller (operating at 10MHz) can accurately determine the delay. The sensor connector comprises of

power pins, a trigger-input pin, and an echo-detect output pin. The trigger-input pin is connected to an output latch that is addressable by the microcontroller. The echo-detect output pin is connected to one of the interrupt pins on the microcontroller. This allows the microcontroller to work in parallel with the sensor, only having to respond when an interrupt is raised. According to the spec sheet, it takes 58uS of time for the “ping” to reflect off an object 1cm away. This interval grows linearly with distance. The sensor is accurate for objects up to 3m away. For this project, 3m is more than enough range.

Infrared Emitter / Detector

The infrared emitter / detector comprise of a simple infrared LED and an infrared photosensitive chip. The infrared emitter is a two-terminal LED with a max current of 30mA. It will be connected through a resistor to 5V. The infrared detector has two power pins and a sensor pin that is connected to one of the input pins on the microcontroller. The detector performs all of the amplifying of the phototransistor internally and merely outputs a high or low digital signal. As this signal may become erroneous from sunlight and other IR sources, it is not connected to an interrupt pin and is merely sampled at a slow-rate where it is then de-bounced. In order to prevent problems from erroneous signals, the microcontroller can be set to ignore this sensor’s input.

Wireless Transceiver

There are two wireless transceiver sensors in this system's design. One is located on the wireless bridge connected to the computer, and the other is on the car. The two sensors act as endpoints in a wireless link allowing bidirectional communication between the computer and the car. The wireless transceiver used in this project is the "RF-24G". It operates at 3.3V, has an internal antenna, and works at frequencies around 2.4GHz. It transmits/receives data at up to 1Mbps providing more than enough bandwidth for this project. As the chip operates at 3.3V and the microcontroller operates at 5V, all of the pins go through a level shifter chip to change the voltage accordingly. The level shifter used for all but two of the pins is an 8-bit voltage level shifter chip "SN74LVC4245A". The remaining two pins use a 1-bit voltage level shifter chip "SN74LVC1T45". Only one of these pins is bidirectional. It uses the 1-bit voltage level shifter chip as it has the added feature of being a transceiver. Communication with the wireless transceiver is preformed serially utilizing a microcontroller-driven clock and a bidirectional data pin. The wireless transceiver data and clock pins are connected to the "Master Synchronous Serial Port" (MSSP) subsystem of the microcontroller. As the MSSP uses two data pins and the wireless transceiver uses only one, the two data pins are connected together. This requires the code to set the output pin to tri-state when wanting to receive from the transceiver. Utilizing a microcontroller subsystem allows communication with the transceiver to occur in parallel with the main microcontroller instruction flow. Additionally, interrupts are used to notify the

microcontroller when data is received or when outgoing data has been sent. The other aspects of the transceiver's protocol are too lengthy to attach to this report, and can be found in reference manuals available online.

Wireless Color Video Camera

The wireless color video camera is the main sensor of the autonomous driving system. It comprises of the camera itself mounted on the front of the RC car and a receiving unit connected to a capture device connected to the computer. The camera and receiving unit are purchased as a pair and is manufactured by "Astak" with part # "CAM-WL809". The video it transmits is around TV-quality. The receiving station outputs the signal using RCA which is then read by a "KWORLD Xpert DVD Maker USB 2.0 (VS-USB 2800)" capture device that is installed on the computer. The capture device is supported by Windows XP and allows other programs to read the video stream.

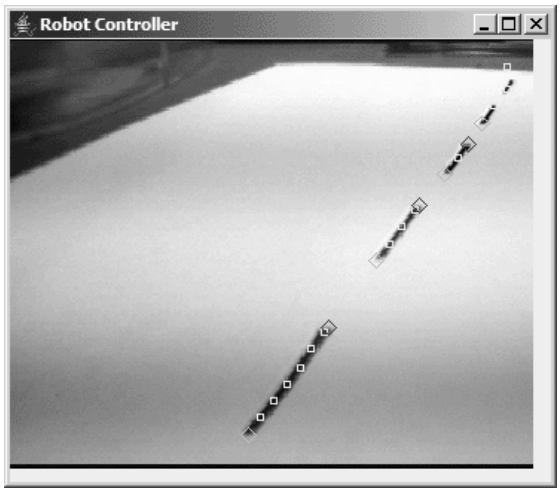
All of the processing for the video stream and majority of the "thinking" for the car are performed in a monolithic Java program I am creating. Java was chosen for its extensive support and extremely powerful features. It was a concern initially that Java may not be fast enough to perform the video processing in real-time. However, this has not proven to be the case. At the time of writing this, the core video processing routines are complete and working exceptionally. The routines have the ability to find lines drawn on a piece of paper and trace their path. Additionally, it can decipher and trace dashed lines

correctly. All of these operations can complete successful at nearly 15 frames per second.

The software code was written entirely by me and is on going constant revision. It currently spans several hundred lines and unfortunately is too large to include in this report; however, it is available upon request.

The basic software design relies on two threads working in conjunction. The first thread performs all the operations necessary to retrieve and display the video stream from the capture device. All communication with the capture device is performed through the standardized Java Media Framework interfaces, allowing the code to be capture device independent. The thread operates at the frame-rate where it retrieves the display buffer from the capture device. It then gives the second thread a chance to manipulate the display and then displays the image on the screen. The second thread is an endless loop of retrieving a copy of the display buffer, performing its line tracing algorithms, and then generating a visual interpretation that the first thread displays at the next frame update. This architecture is used as the line tracing algorithms could potentially take longer to perform than the frame rate updates. With this architecture, the image on the screen is always the newest and the visual interpretation is accurate as of the last frame the algorithms processed. In practice, the algorithms are running fast enough that the visual representation is delayed by only one frame.

Included below are screen shots demonstrating that the algorithm is running successfully. For demonstration purposes, a sheet of white printer paper was drawn on by a black magic marker. The lines are made to mimic road lines.



On the left screen shot, a paper with a straight line drawn on it is placed in front of the camera. The algorithm correctly identifies the line and traces it up its length. On the right screen shot a dashed line drawing is placed in front of the camera. The camera correctly accommodates the dashed nature and even associates the dashed parts together as all the same line. For either screen shot, the algorithm worked regardless of the orientation of the line and worked well even if the paper was rapidly shifted from side to side.

The visual interpretation symbols are as follows: A gray diamond is placed at the beginning of the line. The beginning of the line does not have to be at the bottom of the camera's field of view. It then symbolizes tracing the line by placing periodic yellow squares every so many pixels on top of the line. If the end of the line is reached, a red diamond is placed on top of the end of the line.

If it is a dashed-line, a green diamond is placed on the beginning of the next dash, marking the continuation of a line, and the process continues for its length.

When the software is complete, the interpretation of the line will then be passed to another subsystem to perform the geometric calculations to estimate the distance and curves of the lines. This will be used by the driving algorithms to correctly steer the car. The ability of the visual algorithm to correctly interpret dashed lines provides the added ability for the car to determine how fast it's moving purely from the video stream. The logic will be added to track the movements of the road dashes from frame to frame. This can be used with knowledge of the frame rate and distance calculations to determine the speed of the car.

Conclusion

The sensors for this project are extremely varied and complex. Initially, it was the wireless video camera sensor that I believe would cause me trouble. However, with the testing nearly complete, the camera has exceeded all expectations. The ultrasound range finder uses a very simple interface and I do not expect it to cause me any problems. The IR sensors work perfectly. The last sensor to bring-up is the wireless transceiver. It is an extremely complicated chip and I have been performing extensive research of it. I have almost completed writing the software to control it and it will be working soon.