Autonomous Car
Group Number 11
Embedded System Lab Project

Project Report

April 15, 2011

By
Hemant Noval 08005017
Palak Dalal 08005034
Sushil Meena 08005014
Mudit Malpani 08005020

Under the Guidance of
Prof. Kavi Arya

Department of Computer Science
Indian Institute of Technology Bombay
Powai-400076

April 2011

Abstract

We will make prototype of an Autonomous Bot using FIREBIRD V robot. This bot will move throgh the road towards its destination automatically . While moving though the road it will also follow the traffic signals ,speed limit and lane system for each road.The overhead camera (acting as GPS) will take photograph of the arena (road network) and will make an internal represenation of road network and will also identify the postion of bot on the road.Bot will use IR sensors and Sharp sensor to detect vehicle presence in front of it thus avoiding collision.Optical Character recognition algorithm will be used to read speed limit sign boards. Djikstra Algorithm is used to find the shortest path from source to destination.

Contents

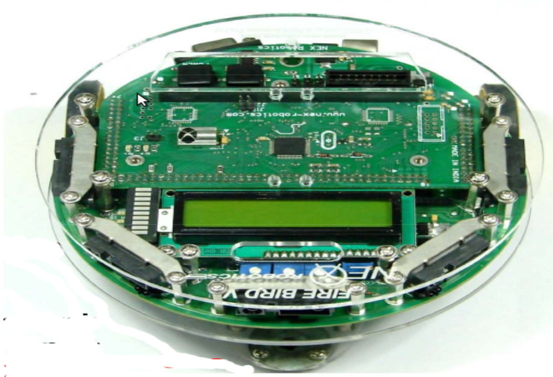
1 Introduction

Our autonomous car moves on the road following the traffic signals and sign boards. The driver specifies the destination, and the central server calculates the path to be taken which the car bot follows and communicates it to the car.The car follows the path maintaining the lane all-throughout even at turns. The car is able to read and process traffic signs specifying acceptable speeds using OCR for speed control.

2 System Architecture

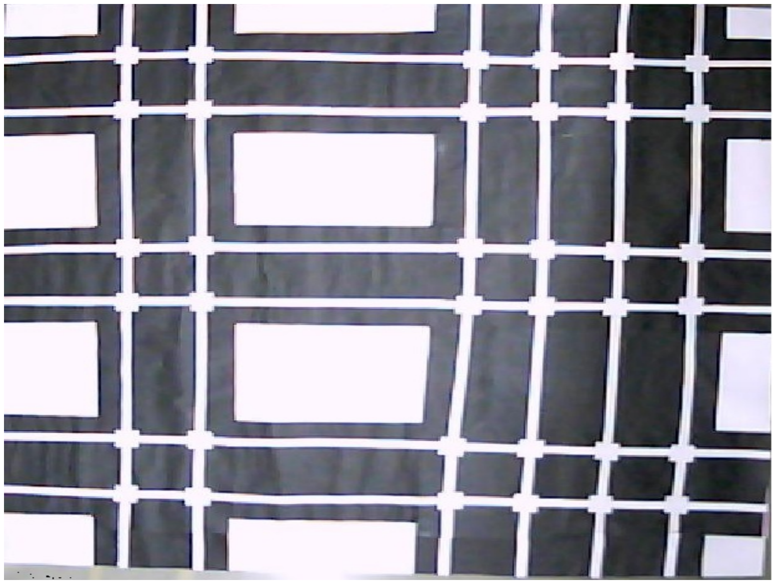
2.1 FireBird

In our prototype FireBird-V robot will be used as a car



2.2 Road Network arena

The arean is a network of black roads, each road can have any even number of white lanes. The crossings between any two lanes is marked further by pasting a small square over it. Each road follows the lane system, ie. half of the lanes on a road are for forward moving traffic and other half for backward moving traffic. (This information is used in finding the path to the destination) Here is a sample arena (6ft x 6ft), which we used for demo and testing - The arena consists of one four lane road and four two-lane roads as shown in figure.



2.3 Overhead Camera

An overhead SLR camera is setup over the arena to detect the road network and position of car on the map. This simulates a GPS module. The camera we used was of 12 Mega Pixels.

2.4 Camera on Car

A SLR camera has to be setup on the car to read road signs. The orientation of camera should be proper so that the signs are in its field of view. We placed the signs to the left of the road and camera is turned to left to view the signs.

2.5 Zigbee Module

Each car will have one in vehicle unit which is ZigBee enabled transreceiver module, which will communicate to the central server.

2.6 Central Server

A laptop is connected to the overhead camera as the central server. It processes the image of arena. This laptop will also be connected to the on-board camera and thus will do the required image processing of traffic signals and speed limit sign boards.

3 Overall Description

The main idea of this project is to develop an autonomous system of cars that can be guided by GPS to reach the destination. The driver specifies the destination where he wants to go. The GPS would find a path to the destination. The car follows the path maintaining the lane all-throughout even at turns. The car is able to read and process traffic signs specifying acceptable speeds using OCR for speed control. It also finds application in slowing down on seeing a speed breaker sign. In case, the speed limit is exceeded, an SMS is sent to the car driver to slow down. A fine can be imposed in case there is a large speed limit violation. It will also implement stop and go at service lanes by displaying appropriate signs.

User is person with knowledge of the place where he wants to reach. The car automatically takes him there using GPS and while maintaining acceptable speed.

The main constraints include detecting exact position of the car using GPS, maintaining lanes during turns and exact speed control.

Assumptions include presence of appropriate traffic signs on the path. To implement stop and go at service lanes we are displaying appropriate signs to the autonomous car. We are assuming that traffic signs are in the field of view of camera on the autonomous car.

4 Details

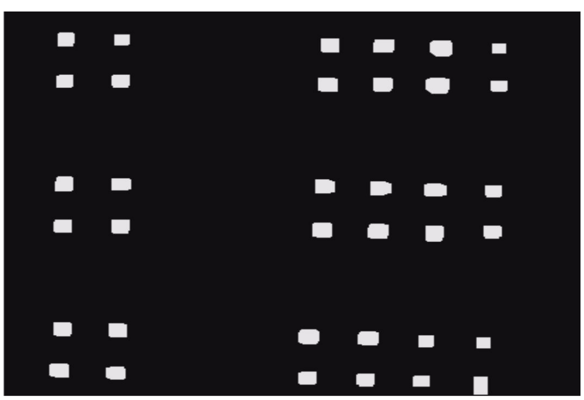
4.1 Functionality

4.1.1 Road Network Detection

It takes image of the arena and makes road network out of it. The image contains black roads with white lines over it and junction have slightly bigger white blobs. There are several color parameters that need to be configured. These are placed below under config variables heading. Cropping needs to be done to remove outside arena objects. Finally two files are generated-one contains the junction coordinates and second contains the edge information.

Algorithm:

1. Captured image is cropped to remove side noise.
2. Smoothening is applied to remove even further noise.
3. Then using erosion followed by dilation and subtraction from original image, very big white areas(places between roads) are removed to get image having only road network (orgImgWithoutBigWhite).
4. Image having only nodes is acheived by carefully eroding and dilating road network(orgImgWithoutBigWhite)using structure element structElemRemovRoad.
5. Connected components of this image is taken. The junctions in road network are given special color value 100.
6. For each node flood fill is called. This gives its connectivity to all other junction. Using this information appropriate information is filled in the output files



4.1.2 Detection of start position of car

It takes image of car on the arena from an overhead camera and finds its coordinates and orientation. The car has two tags on it - a red tag towards the forward direction and a blue tag towards the backward direction. This code finds the centres of the two tags and puts them in file "botPos".

Algorithm:

1. It first identifies red and blue colors in the image, makes two matrices containing red and blue colors respectively.
2. It finds connected components in each of the two matrices, rejects the ones not in the specified range of area.
3. The remaining connected components are the required ones. It finds their centres and put in the file.

4.1.3 Path Detection

Combining road junction It combines the lane junctions of the same road junction into a single node and creates the adjecency matrix for these new nodes/road junctions. It also calculates the source node i.e. the node nearest to the car bot.

Algorithm.

1. The lane junctions obtained through image processing are combined using the distance between them. If the distance is lerss than a particular threshold the are combined to form a single node.
2. The coordinates of new nodes/road junctions are the average of the corresponding lane junctions. This will approximately be the coordinates of the midpoint of the road junction.
3. An adjacency matrix is formed where each entry represents the number of lanes on the road connecting the two nodes.
4. To find the source node, first all nodes on the road on which bot is currently standing are calculated.
5. The node closest to the bot, in forward direction is the source node.

4.1.4 Shortest path

The driver specifies the destination where he wishes to go through command line interface. A path is found using the map.

Algorithm.

1. The graph is formed using the road junctions as nodes and roads as edges with weights equal to the length of road i.e. distance between the two nodes.

2. The shortest path between the source node, identified through gps and the destination node, provided by user is calculated using dijkstra algorithm.

3. At each node to be travelled by user, three things are calculated:

   (a) Directions: To reach the next node in which direction should bot turn. Calculated using the slopes of the current road and the next road(to identify whether to turn or go straight) and the coordinates of the next node(to identify left/right turn).

   (b) Lanes skipped before turn: Bot needs to skip a few lanes before turning according to the laning system. Calculated using number of lanes on the perpendicular road.

   (c) Lanes to be skipped after turn: On right turn, few lanes of the previous road need to be skipped before starting to move on the new road. Calculated using the lanes on current(before turning) road.

4. These three information for every node is transmitted to the bot thorugh zigbee

4.1.5 Traffc Symbol reading. functioning:

It captures images from bot camera, processes it and sends encoded message to bot using zigbee module about what it has seen- red light, green light, speed limits (30,60,90) or nothing special.

Algorithm:

1. Captures image and calls traffic light module, extracting character image and OCR modules in sequence.

2. Traffic light module:

   (a) Image is input.

   (b) Red pixels (red color range is configurable), is extracted and turned into white and other pixels black.

   (c) Smoothening done to remove unwanted small red pixels.

   (d) Connected components are extracted from image.

   (e) Those components whose size are within a given configurable area range, are qualified as red light.

   (f) Return code set to 1.

   (g) Same thing done for Green and if green light found, code is turned to 2.

   (h) Else 0 is returned.

3. Extracting character Image:

   (a) Image is input.

(b) Blue pixels extracted from image and turned white. Rest all are black.

(c) Using dilation and erosions, image containing only the circle is extracted. Circle is white and other pixels are black.

(d) Pixels that lie outside the circle in original image (corresponding to black pixels in binary Circle image), are turned white.

(e) Image is turned into binary image, after some smoothening and the resulting image is the image that has only the characters.

(f) Inverse of this image is returned.

4. OCR:

   (a). Above image is fed to OCR module, which gives the resulting characters in form of an array.

5. Zigbee communication:

   (a) Codes assigned : a-¿ red light , b-¿ green light , c-¿ number 30 , d-¿number 60 , e-¿ number 90

   (b) Depending on the image that has been identified, above codes are communicated to the bot using Zigbee module.

4.1.6 Image Processing

Whenever a turn is to be taken, the amount by which to turn to maintain the correct lane is determined by image processing of divider and distance from it. The stop and go signs at service lanes are read using image processing.

4.2 Working Specifications

4.2.1 Will follow

1. Traffic signals at junctions i.e will stop if signal is red and move if signal is green.
2. Speed limit boards at appropriate places.
3. Speed Breaker boards and slow down appropriately.
4. Zebra crossing boards

4.2.2 Lane following and collision avoidance

Will move between two white lines (on a black surface) which will be analogous to the lane system implemented in real road netwroks so as to differentiate between fast and slow moving traffic. This will also avoid collisions among cars (bots) whose difference in speeds is very high. Will slow down or stop if there is an obstacle ahead depending on the movement of the object.

4.3 Supportibility

1. The code should be written in a modular fashion.
2. Interfaces of each module should be properly defined.
3. Standard naming conventions should be followed for variables and functions.
4. In addition these variable and function names should be self explanatory.
5. The code should be well commented.

4.4 Design Constraints

1. Precision in calculation of speed
2. Low resolution of camera.
3. Maintaining lane while turning.

4.5 On-line User Documentation and Help System Requirements

1. Detailed Documentation
2. Final Report

4.6 Interfaces

The system requires the following interfaces for its functionality.

4.6.1 User Interfaces

Command Interface for user to specify the destination.

4.6.2 Hardware Interfaces

1. A Fire-Bird robot which will act as the autonomous car.
2. 3-4 smaller robots for traffic.
3. A global camera interface to view whole arena and identify the current position of the car.
4. Camera interface on the car bot to view the traffic lights, sign boards and speed-breaker
5. ZigBee interface at both robots and the central server to communicate the path to be followed.

4.6.3 Software Interfaces

1. MATLAB
2. WinAVR
3. AVR Studio
4. GCC Compiler

4.6.4 Communications Interfaces

1. ZigBee
2. GSM Module

5 Work Division

1. Line Following and all the code burnt on bot : Hemant and Mudit
2. Bot detection and Map Detection : Palak and Sushil
3. Shortest Path : Hemant and Mudit
4. Optical Character recognition : Sushil and Palak
5. Arena : All four

6 Project plan

1. Line following : 16th March
2. Map Detection : 20th March
3. Bot Detection : 21st March
4. Optical Character Recognition : 25th March
5. Djisktra Algorithm : 28th March
6. Zigbee communication : 29th March
7. Prototype Demo : 30th March
8. Arena : 31st March to 3rd April
9. Integration : 31st March to 6th April