Autonomous Robot using Computer vision

Aditya Godambe
K.J. Somaiya College of Engineering, Vidyavihar
Mumbai 400077
aditya.godambe@somaiya.edu

Ashutosh Mahajan
K.J. Somaiya College of Engineering, Vidyavihar
Mumbai 400077
ashutosh.mahajan@somaiya.edu

Chinmay Karanjkar
K.J. Somaiya College of Engineering, Vidyavihar
Mumbai 400077
c.karanjkar@somaiya.edu

Poonam Bhogle
K.J. Somaiya College of Engineering, Vidyavihar
Mumbai 400077
poonambhogle@somaiya.edu

ABSTRACT

Autonomous Machines are now becoming a major part of the industrial and commercial world. In this paper, we discuss one such application viz. Driver-less or autonomous mobile vehicles. While Google Driverless cars use LIDAR technology for environment sensing, this project provides an alternative using Image Processing, Computer Vision and Machine Learning. It presents a working model of an autonomous robot which uses Computer Vision and Machine Learning for Obstacle Classification. Machine Learning is used for training of the dataset of various obstacles types. Computer Vision is used to grab and process the video images captured by the web camera eye of the robot. Image processing is used for other image operations like detecting moving obstacles, colors, signs prevailing in the surrounding of the robot. Sensors (Infrared and Ultrasonic) are used for navigation of the robot. The Hardware backbone of this prototype is built up using simple microcontroller, sensors and motor drivers that are used for relaying the information between hardware and software.

General Terms

Image Processing, Autonomous vehicle

Keywords

Image Processing, Autonomous vehicle, MATLAB, Arduino, Computer Vision, Machine Learning, Image Classification

1. INTRODUCTION

Autonomous Machines are machines that are capable of sensing their environment and operating without human interaction. The sensory elements include different sensors right from simple range sensors to complex circuit embedded camera systems and olfactory sensing integrated modules. Various microcontroller and programmable electronic chip boards are used to enable the functionality of these machines. [1]

Autonomous Vehicle is one such application of self-governing machines. An intelligent robot capable of taking navigation decisions on its own, classifying the obstacles, recognizing road signs and blocks, signals fall in the scope of this paper. In general, autonomous vehicles use a variety of techniques to detect their surroundings, such as radar, laser light, GPS, Odometry, and computer vision. Already existing projects such as the Google Driverless Cars uses LIDAR (Light Detection and Ranging) technology. [2]

The prototype on which this paper is based uses a scaled down technology like Image Processing, Machine Learning, Computer Vision as a part of software technologies and Sonic Wave Range Detection and simple LED Line Following module as a part of hardware technologies. The Image Processing and Range Detection modules of the robot explained in this project enables it to analyze the visual data received from the sensors. The line following module helps in keeping the robotic vehicle in its path while it avoids the obstacle and turn around them. A detailed explanation of the methodology used for building the robot is explained in further sections.

Autonomous Robots have a wide range of applications in domestic as well as industrial operations. Ranging from simple material transport operation to high security military ranging operations autonomous vehicles find itself useful in high importance jobs like ranging and spying. The technology developed in this project can be further expanded in its scope and style of functionality to provide efficient and optimized performance along with extended features. These features include creating an embedded hardware system, use of high precision image capturing sensors, fast microcontrollers and software tools to provide improved time and space complexity along with improved practical performance. These features can be further expanded to connecting the robot with wireless technologies for fast data (audio, video) transmission and enabling GPS (for domestic transport) or map schematics (for military applications) for source to destination and accurate navigation.

2. EXISTING AUTONOMOUS TECHNOLOGIES

This project was inspired from the famous Google Driverless Car project. It aims at providing a scaled down non-superfluous approach to this particular concept of Autonomous Vehicles. Along with this project various other projects based on the similar concept have been in progress taken up by various university organizations like Stanford and other companies like Audi, Apple, BMW, Uber, Volvo.

Basic requirement that are addressed during the construction of these technologies are Self-Maintenance, Sensing Environment, Task performance, Autonomous Navigation (Indoor and Outdoor). Technologies like Artificial Intelligence, Computer Vision, Neural Networks, Machine Learning, Imaging Processing, Embedded Systems. General Technologies used involve sensors and are inclined towards electronic approach. Advanced and modern technologies like Artificial Intelligence and Computer Vision provide a more strong and efficient approach towards this concept. [4]

The Google Driverless Car project (now called Waymo) work on Light Detection and Ranging Technology. For the Vehicle part of the project Google has developed their own custom vehicle assembled by Roush Enterprises with parts supplied from Bosch, ZF Lenksysteme, LG, and Continental. [2] The technologies used for the project involve Laser Range Finder, Front Camera for near vision, Bumper mounted Radar, Aerial Module for reading precise Geo-Location, Ultra-sonic sensors for rear wheels, devices like altimeters, gyroscopes, and tachymeters that determine the very precise position of the car, synergistic combination of sensors for collection and relaying of data to the CPU of the car. The LIDAR creates a virtual map of the cars environment which is then analyzed to provide appropriate decisions while driving. Successful On-Road tests have been conducted by the project team to demonstrate and verify the working of the project.

3. SYSTEM ARCHITECHTURE

The given diagram depicts the distribution of the various modules of the system. The flow of control and processing is shown via the interconnecting arrows between the modules which basically explains how the system processes the input image, identifies the object present in it and forwards the robotic operations via the controller to the hardware.

The system is broadly classified into two major working areas as the Hardware and Software. Different modules are then designed under these areas. An interfacing link is created between the two areas to allow communication and flow of data.

3.1.2 *Obstacle detection*

This module analyses each of the frames split from the video feed and detects and identifies the obstacle using the classification function. This

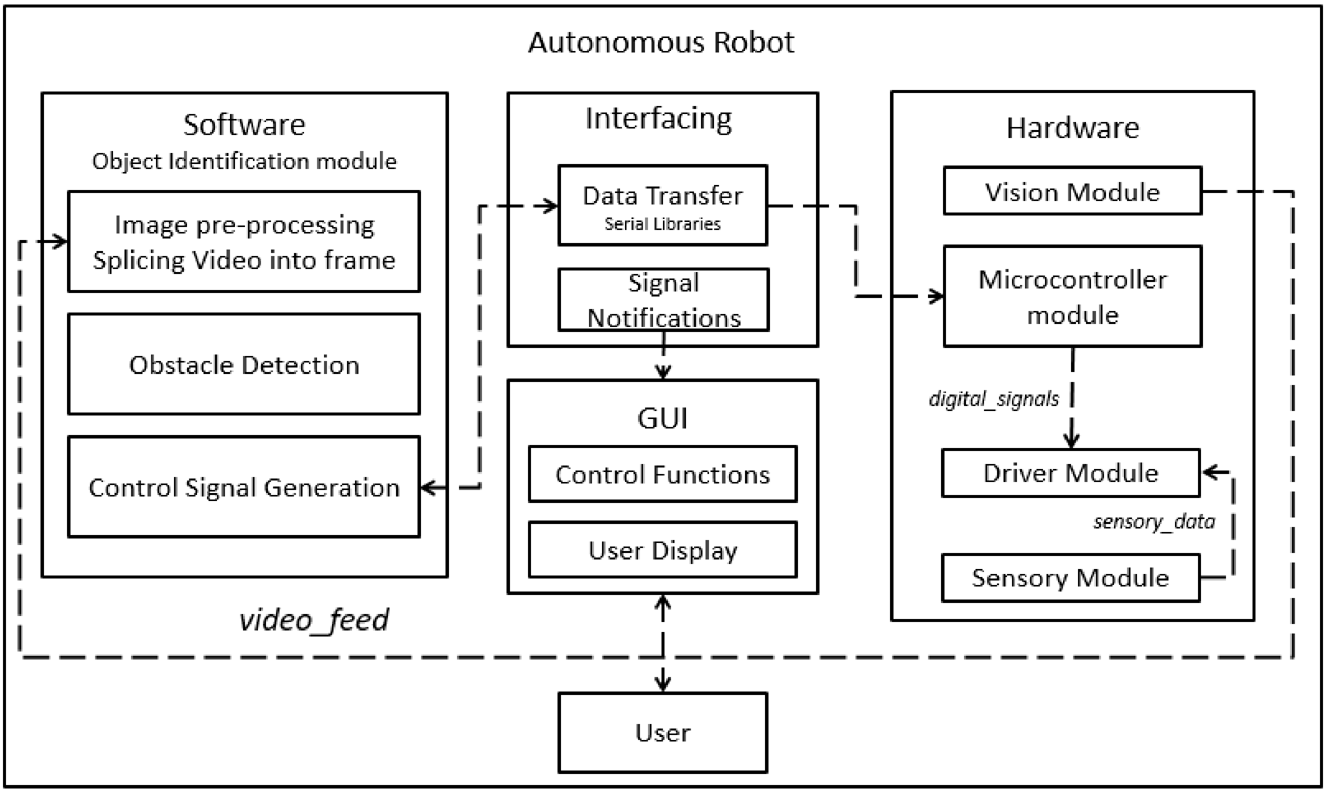


Figure 1: System Architecture

The GUI controller will display real time data to the user and is included as a module for direct human interaction limited to only monitoring of the system and not involving computing interaction.

classification function compares the obstacle with derived feature vectors and identifies it on the closest match.

3.1.3 *Control signal generation*

Depending on the type of obstacle identified, this module sends appropriate data(flags) to the Interfacing layer which converts them and sends them to the microcontroller.

3.1 The software layer

3.1.1 *Image pre-processing*

This module directly receives the video feed of the surroundings from the Vision module. This video feed is split into multiple frames at some intervals and forwarded to the obstacle detection module.

3.2 The Interfacing layer

3.2.1 *Data Transfer*

This module receives control signals from the Control signal generation module and converts them into Serial data which can be understood by the microcontroller using Serial libraries. It communicates with the hardware via a Serial port.

3.2.2 *Signal notifications*

Upon classification of the obstacle, the category is sent to the User Display via this module. This module notifies the user about the output of the obstacle identification via the GUI.

3.3 The Hardware layer

3.3.1 *Vision module*

This module receives the video feed from the web camera attached to the robot. The primary function of this module is to receive input about the surrounding area from the camera and forward it to the Image pre – processing module.

3.3.2 *Microcontroller module*

This is the most important module in the system as it governs the working of the robot, decides its movement based on the control signals received and coordinates with all of the other hardware module using digital signals.

3.3.3 *Sensory module*

This module receives input from the line sensors which are used to guide the robot on a pre-defined path and range sensors which are used to determine the distance between the robot and the surrounding obstacle. This sensory input is sent to the Driver module for navigation and control.

3.3.4 *Driver module*

This module controls the motors of the robot and decides the motion, direction and speed. It receives input from the sensory module which notifies it about the path and the existence of surroundings obstacles.

3.4 The GUI layer

3.4.1 *Control functions*

This module defines several control functions which allow the user to control the bot's movement in case human intervention is required. This module is necessary as it ensures manual control over the robot in case of emergencies such as battery failure, damage to the robot and error in processing.

3.4.2 *User display*

This module displays the video feed of the web camera mounted on the robot.

4. SOFTWARE IMPLEMENTATION

4.1 Signal Detection Module

- Extracting the video from web cam and opening port to Arduino
- Grab the video and set frame and color properties
- Enter the loop and subtract red from the RGB format and store it in an array after converting it into binary and after removing pixels less than 300px.
- Classify the red pixels and send signal to Arduino accordingly.

4.2 Image Categorization and Identification[7]

4.2.1 *imageDatastore*

imageDatastore(location) creates a datastore from the collection of image data specified by location. A datastore is a repository for collections of data that are too large to fit in memory. After creating an ImageDatastore object, you can read and process the data in various ways.

4.2.2 *bagOfFeatures*

bagOfFeatures(imds) returns a bag of features object. The bag output object is generated using samples from the ImageDataStore object as input. The

ImageDataStore object given as input is used to by the bagOfFeatures object to extract features and thus define feature vectors. It defines the features, or visual words, by using the k-means clustering (Statistics and Machine Learning Toolbox) algorithm on the feature descriptors extracted from trainingSets.

4.2.3 *trainImageCategoryClassifier*

trainImageCategoryClassifier(imds,bag) returns an image category classifier. The classifier contains the number of categories and the category labels for the input imds images. It accepts an ImageDataStore object and a bagOfFeatures object as paramateres. The function trains a support vector machine (SVM) multiclass classifier using the input bag, a bagOfFeatures object.

4.3 K-means clustering algorithm implementation [7]

K-means clustering algorithm is primarily used for partitioning the data into K number of clusters which are mutually exclusive. It returns the index of that cluster which matches to the assigned observation. [9]

K-means partitions the data by clustering the objects within the data which are close to each other and separating those which are different from each other into different clusters. Each cluster has its cluster center and member object which uniquely defines it. The center is calculated as the point where the sum of the distances from all objects in that cluster is the least.

The reduction of the number of features [7] through quantization of feature space is done using K-means clustering algorithm. It is done as follows:

1. The image's features are first detected and subsequently converted into feature vectors.
2. As there are hundreds of images in the datasets, the number of descriptors collected are large. K-means clustering algorithm iteratively groups the descriptors into k mutually exclusive clusters. The resulting clusters are compact and separated by similar characteristics.
3. Each cluster center represents a feature, or visual word.

5. HARDWARE IMPLEMENTATION

5.1 Importance of Hardware.

For any software application to work hardware becomes an essential component to provide a platform. In this project, the main hardware will include all the necessary components required to build a simple robotic vehicle capable of processing the required modules. Actuators and sensors (used in reference to human machine interaction) are used to enable the robot to interact with the environment.

5.2 List of Major components

- Laptop (Processor)
- Arduino UNO
- Web camera
- L293D Motor Driver
- 60rpm Motors
- HC SR-04 range sensors
- 4.8V Servo Motor
- 12V battery Supply
- Breadboard and Connectors
- Metal Frame Chassis
- Resistors, Diodes

5.3 Significance of Parts explained

5.3.1 Laptop

This is the main processing unit of the robot. All the Image processing and Machine Learning Algorithms are run on the laptop. The Video input taken from the camera is relayed to the Laptop which processes it and then provides the suitable signals required for running the robot.

Software processing tools like MATLAB and Arduino IDE are executed on the Laptop. The entire code generation is done here.

5.3.2 Arduino

The main use of Arduino in this project is to take input from the web camera pass it to the software (Image Processing and Machine Learning) module and relay back the output from software to the motor driver and the servo motor.

5.3.3 Web Camera

An external web camera works as the eye of the autonomous robot. It captures a video image of the surrounding and passes it to the main system for processing. It captures and records a video of 640*480 resolution using a 30 Megapixel interpolated lens and CMOS sensor.

5.3.4 L293D Motor Driver [8]

The L293D motor driver help run DC motors in both directions. It is controlled by Arduino ports and is programmed through it. The module has its significance because DC motors cannot be directly programmed and hence cannot be controlled directly by the Arduino.

5.3.6 *Motors*

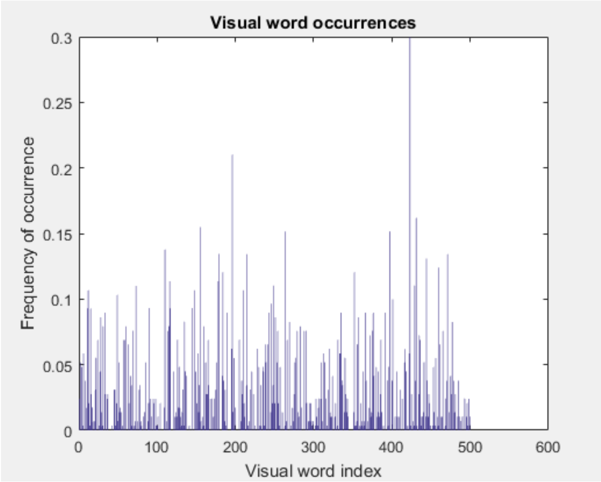
The robot is powered by two 60 rpm motors controlled by the motor driver and works on 12 V of supply. Though the speed of the motor is moderate the power provided by it sufficient to carry the load of the entire robot.

5.3.7 Servo Motor

Servo motors are generally high performance stepper motors and provide rotatory functions. The S3003 servo motor is used in this project to enable the turning of the robot. The front axle of the robot is connected to the servo motor and provide sufficient angle for the robot to turn allowing it to change its linear direction.

5.3.8 HC SR-04 Range Sensors

HCSR04 is a range sensing module that provides 2cm to 400cm of non-contact range measurement with an average



accuracy of 3mm. The range sensors help in analyzing the environment while the bot turns around an obstacle. The sensors are placed sideways and will give signal upon extremely close proximity, while avoid unwanted crashing.

5.3.9 Battery Supply

Used to provide the running power to the robot, since no external supply can be connected.

5.4 Arduino MATLAB Interfacing

The main interfacing between hardware and software is provided through this system integration. Arduino provides the Serial Library that allows code integration and data communication between MATLAB and Arduino IDE. The data received by the chip can thus be directly passed to MATLAB for processing. This data that flows initially into MATLAB comes from the web camera. Upon processing (explained in the software section) the software thus sends appropriate signals. These signals are then relayed through Arduino chip and motor driver that allows required running and stopping of the DC motors. A separate communication is provided to control the servo motor.

6. FLOW OF CONTROL [6]

1. The primary source of input to the processing unit are the images captured by the webcam which has been mounted on the laptop as mentioned above. The location of these images is given to the MATLAB file for importing the images for analysis.
2. The MATLAB file is connected to a large dataset containing images of possible obstacles the robot might encounter. These image sets contain large collections of images of the obstacles shot from every direction.
3. The MATLAB file then splits the image sets into training and testing data. The training data is used to train the classifier while the testing data is used to measure the accuracy of the classifier's working.
4. After splitting the data, the features are extracted from all the image categories and then used to construct the visual vocabulary. Encoded training images from each category are fed into a classifier training process.
5. We then evaluate the classifier's performance by using the evaluate function. The evaluate function returns the confusion matrix, which is a good initial indicator of how well the classifier is performing.
6. Thus the classifier has been trained for image recognition. This classifier is used to recognize the obstacles present in the environment. Upon recognition, the processing unit sends appropriate instructions to the Arduino chip which forwards it to the motors.
7. The above communication decides the robot's movement and reaction to the obstacle. Depending on the category, the Arduino is further instructed to move accordingly. For example: If the object has been classified as a paper ball, no explicit movements are done by the robot as a paper ball does not hinder the motion of the bot. If the object has been classified as a tree/vehicle, then the robot is given instructions to avoid the obstacle.

Figure 2: Histogram of visual word occurrences in an image

The above histogram depicts the comparison between the frequency of the visual word occurrences in a given image set as opposed to the visual word index. It can be described as a graphical representation of the feature vector of each image. The y-axis shows the frequency of occurrences of the visual words while the x-axis shows the visual word index (no of visual words used for clustering).

It forms a basis for training a classifier and for the actual image classification. This tell us that higher frequency of more number of visual words would lead to better matching of the input test image and training dataset; which eventually would increase the efficiency of the classifier as it would have more number of visual words to compare so as to assign an image to a particular category. For better prediction of the image into a particular category tall and closely spaced bars of histogram are required and the training and validation dataset set should be designed to get the optimized histogram representation.

7. ACKNOWLEDGMENTS

This research was supported by K.J. Somaiya College of Engineering, Mumbai. We would thank Ms.Suchita Patil for assistance with the project methodology, and Ms.Grishma Sharma for comments that greatly improved the manuscript.

8. CONCLUSION

In this paper, we proposed a new method for building autonomous vehicles specifically designed for small scale applications. We have explained the technology required to build a prototype model which integrates Computer Vision and Machine learning techniques which enables the vehicle to run autonomously. The major advantage of this technology is the small scale and relatively easy process which can be easily implemented. Efficiency measurements for the image classification algorithm were done to reveal that the efficiency is based on image feature parameters. Average accuracy of 64% (with lowest feature parameters) and 77%(with highest feature parameters). However, the time complexity increasing as the efficiency is improved and due to the increase in size of training and validation datasets.

Further work can be carried over the existing prototype to give maximum efficiency and optimized working as well as the design of the robotic vehicle. Features like GPS navigation and use of algorithms like background subtraction(for mobile obstacles) can be used to provide additional functionality and improvised performance to the robot.

9. REFERENCES

[1] J. S. Albus and A. M. Meystel, Engineering of Mind: An Introduction to the Science of Intelligent Systems, John Wiley & Sons, New York, (2001).

[2]"Autonomous car", https://en.wikipedia.org/wiki/Autonomous_car

[3]Al-Zahee, T.S.A, et. al., Lane Tracking and Obstacle Avoidance for Autonomous Ground Vehicle, 7th Europe-Asia Congress on Research and Education in Mechatronics, November 21-23, (2012).

[4] Bush F.N, Esposito J.M, Vision-based lane detection for an autonomous ground vehicle: a comparative field test, 42nd Southeastern Symposium on Digital Object Identifier, March 7-9, (2010).

[5]Mubashir Ahsan, Aahsan Iqbal "Using Image Processing for Design of an Autonomous
Ground Vehicle", International Journal of Computer Applications (0975 – 8887)
Volume 5 – No 10, January 2014

[6] "Feature detection and extraction", https://in.mathworks.com/products/computer-vision/features.html#feature-detection%252C-extraction%252C-and-matching

[7]https://in.mathworks.com/help/vision/examples/image-category-classification-using-bag-of-features.html

[8]http://www.instructables.com/id/How-to-use-the-L293D-Motor-Driver-Arduino-Tutorial/

[9]https://en.wikipedia.org/wiki/K-means_clustering