# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

DIVISIÓN DE MECATRÓNICA Y TECNOLOGÍAS DE LA INFORMACIÓN


## STEREOVISION FEEDBACK AND FUZZY CONTROL FOR AUTONOMOUS ROBOT NAVIGATION

**THESIS**

PRESENTED AS A PARTIAL REQUISITE

TO OBTAIN THE DEGREE:

MASTER OF SCIENCE IN AUTOMATION

BY:

ARISTEO HERNÁNDEZ MARTÍNEZ

MONTERREY, N.L. DECEMBER OF 2010

# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

DIVISIÓN DE MECATRÓNICA Y TECNOLOGÍAS DE LA INFORMACIÓN



TECNOLÓGICO DE MONTERREY®

## STEREOVISION FEEDBACK AND FUZZY CONTROL FOR AUTONOMOUS ROBOT NAVIGATION

**THESIS**

PRESENTED AS A PARTIAL REQUISITE

TO OBTAIN THE DEGREE:

MASTER OF SCIENCE IN AUTOMATION


BY:

ARISTEO HERNÁNDEZ MARTÍNEZ


MONTERREY, N.L. DECEMBER OF 2010

# Contents

# List of Figures

# List of Tables

# Part I

This thesis presents the project of an autonomous vehicle that follows a trajectory by detecting the road through stereovision processing and it is controlled with the use of fuzzy logic. The vision processing implements edge detection by analyzing lines of pixels. The vision processing determines the error in the actual position and the fuzzy control moves the steering of the car and slows down the velocity if needed. It was implemented in a mobile robot to determine current results.

# 1 Introduction

The main objective of this thesis is the implementation of an autonomous robot that is capable of following a road with the use of a stereocamera. The use of vision in 3D is a trend in many vision applications, and we would like to adopt it for autonomous vehicle navigation.

Road following requires two crucial steps: the road recognition and the control of the speed and steering of the vehicle. To reach controllability in these type of robots it is necessary to detect the road and make the image processing to calculate the manipulations needed to keep the robot in the desired position and path.

It can be said that vision systems are a very important factor in the development of autonomous vehicles because they are the sensors of the robot; therefore, the image processing needs to be done on time in order to detect the road and let the entire system to be controllable.

Through this thesis, various techniques in different areas such as road detection, image processing and fuzzy control theory will be examined in order to achieve the objective.

## 1.1 Motivation

Autonomous mobile robots with image processing using computers are a challenge for the control system due to the demands in computational capacity and also the ability to do that processing as fast as possible to allow controllability of the vehicle. Autonomous road following has been researched through recent years because it could prevent accidents in real life situations and it is a trend in autonomous vehicle development. In general, autonomous navigation is an interesting area to research because it demands real mechatronics: electronics, mechanics, computer programming and control engineering.

## 1.2 Problem Statement and Context

Autonomous mobile robotics builds physical systems that can move purposefully and without human intervention in unmodified enviroments and the development of techniques for autonomous navigation constitutes one of the major trens in the current research on robotics. This trend is motivated by the current gap between the available technology and the new application demands. On one hand, the techniques employed in current industrial robots lack the ability to provide flexibility and autonomy; on the other hand, there is a clear emerging market for truly autonomous robots. Possible applications include intelligent service robots for offices, hospitals, and factory floors, maintenance robots for inaccessible areas, domestic robots for cleaning or entertanment, or autonomous vehicles to help the disabled and the ederly.

The problem to be solved by this thesis is the need of an autonomous vehicle capable of following a road autonomously; therefore, the main components for road traveling need to be controlled: speed and steering. In order to solve this problem, individual

situations have to be considered, such as: ways of detecting the road and/or obstacles, types of controls to be used and the implementation of software and hardware.

## 1.3 Solution Proposal

In order to solve the problem, it has been divided in two main areas: stereovision and fuzzy control.

### 1.3.1 Stereovision

To detect the road and the obstacles, stereovision has been proposed to be the sensor of the vehicle because it does not only detects objects in the image, but also measures distances in the frame.

### 1.3.2 Fuzzy Control

Because the control algorithm is also a fundamental part in road following for a smooth ride and to reduce the error in position, fuzzy logic is pretended to be used as an emulator of human expertise, because it extracts knowledge of expert people and works well with imprecise data.

## 1.4 Main Contributions

The main contribution of this thesis is the algorithm of stereovision implemented to road following. This algorithm, although simple, is practical for the objective need because it allows a fast processing of images, which is a very important for the controllability of the autonomous vehicle. This thesis is also the base for further work, in order to expand the applications and funtionality of this study.

## 1.5 Thesis Structure

This thesis shows the work done in order to solve the problem. The thesis is structured as follows,

In chapter two, the theoretical framework is presented, a summary of concepts that are needed to understand what is done in next chapters.

In chapter three, the state of art is presented, where similar works are presented, which can be the base for this thesis.

In chapter four, the proposal for the solution is presented, it includes the hypothesis of how the objective can be completed as best as possible as far as we are concerned.

In chapter five, the experiments with the autonomous mobile robot are shown according to the objectives and also comparatives with similar works. The results of this experiments is shown as well.

Chapter 6 presents the conclusions of this study.

Chapter 7 proposes further work to be done with this thesis.

The section of appendixes is divided in three: hardware specifications, program description and the description of the experiments carried out.

# 2 Theoretical Framework

This chapter shows the context over which this thesis is based on; in order to understand the objective of this work, the theoretical fundaments have to be studied. It is important to mention the bases of road following for autonomous vehicle navigation which for this thesis are basically divided in: stereo vision, roaddetection and image processing, fuzzy control and real-time systems.

Through this chapter, the concepts of fuzzy control are studied to understand the controllability of the autonomous vehicle. Stereo vision is also a fundamental part of the project, which in this case, is the sensor to detect the road and the boundaries of the car; and in order to sense, an image processing of the vision system has to be done.

## 2.1 Stereo Vision

According to [25], an artificial vision system has:

- Camera, captures the images and trasmit them as electric signals, following rules of exploration.
- Interface, it adapts the electric signals produced by the camera to the computer.
- Software, it allows to analyze the scenes and generates the commands for the robot control autonomously and in real time (Artificial Intelligence).

Figure 1 shows the components of an artificial vision system.

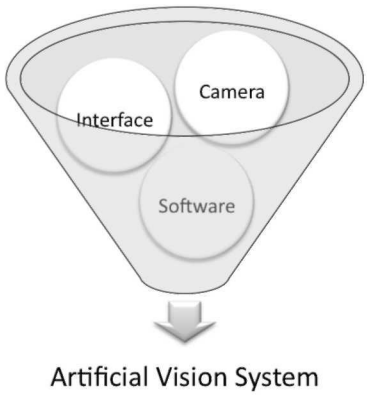


Figure 1: Components of an artificial vision system

In the software part, three consecutive phases can be distinguished:

- Selection, of the useful and indispensable information, because it is almost imposible to take into account all the information the camera provides.
- Interpretation, of the scene in a convenient form for the current application.

- Calculations and generation, of the control orders to the manipulators, according to phase number 2.

Figure 2 shows the components of software for artificial vision system.

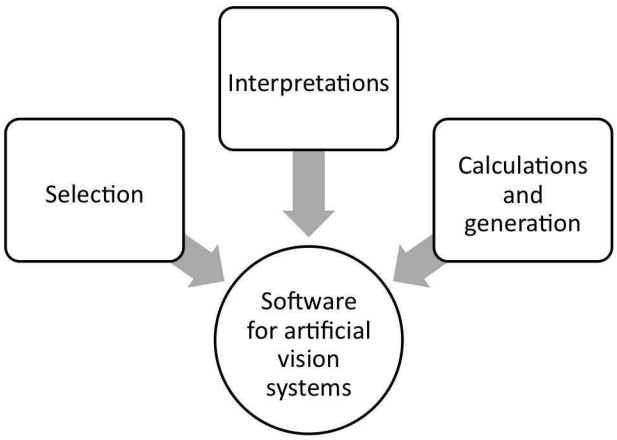


Figure 2: Components of software for artificial vision systems

According to [42], the stereovision based confirmation consists in four major steps:

- determination of regions of interest in the stereoscopic images.
- applicacion of a numerical zoom to maximize the detection range.
- computation of a local disparity map in the regions of interest.
- criterion evaluation from this disparity map to confirm the existence of an obstacle.

Autonomous mobile robots make use of stereo vision to measure their relative distance to obstacles. This method is also relatively inexpensive; while laser scanners can cost tens of thousands of dollars, stereo vision requires only two aligned cameras and some processing power. Stereo vision is a technique that uses two cameras to measure distances from the cameras, similar to human depth perception with human eyes. The process uses two parallel cameras aligned at a known distance of separation. Each camera captures an image and these images are analyzed for common features [12]. Triangulation is used with the relative position of these matched pixels in the images as seen in Figure 3.

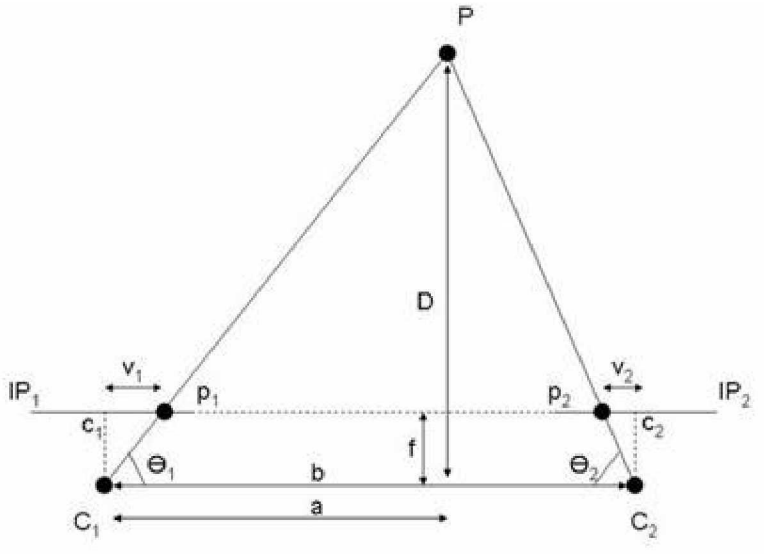


Figure 3: Stereo Vision Triangulation

Triangulation requires knowing the focal length of the camera ($f$), the distance between the camera bases ($b$), and the center of the images on the image plane ($c_1$ and $c_2$). Disparity ($d$) is the difference between the lateral distances to the feature pixel ($v_2$ and $v_1$) on the image plane from their respective centers. Using the concept of similar triangles, the distance from the cameras ($D$) is calculated as

$$D = b\left(\frac{f}{d}\right) \tag{1}$$

The result of the computer vision system is a depth field map which is a grayscale image of the same size to the original image. Each grayscale image represents a distance from the cameras. For example, a white pixel signifies a pixel in the computer's vision near infinity while a black pixel means a point in the infinity distance.

## 2.2 Road Detection and Image Processing

Vision-based road detection is an important research topic in different areas of computer vision such as autonomous driving, car collision warning and pedestrian crossing detection. [3]

Some basic concepts used in image processing are:

- Threshold: is a method of image segmentation. From a grayscale image, thresholding can be used to create binary images. During the process, individual pixels in an image are marked as "object pixels" if their value is greater than some threshold value and as a "background" pixel otherwise.

- Contrast: is the difference in visual properties that makes an object distinguishable from other objects and the background.

- Brightness: is an attribute of visual perception in which a source appears to be radiating or reflecting light. In other words, brightness is the perception elicited by the luminance of a visual target. In the RGB color space, brightness can be thought of as the arithmetic mean $\mu$ of the red, green and blue color coordinates:

$$\mu = \frac{R + G + B}{3} \tag{2}$$

## 2.3 Fuzzy Control

The controllers are the brain of the robot, they do all the necesary calculations so that the robot does what is desired. In this case, the controller is required to maintain the mobile at a certain speed and position. Fuzzy logic was first designed to represent a knowledge expressed in a linguistic or verbal form [21].

### 2.3.1 Definition of the Fuzzy Sets

A way to define a set is to enumerate their elements, and the other, to use a function $P(x)$, where every element $x$ of the set has a property $P$. A third way that is to define a set A using a characteristic function. Let $A$ define the domain $X$. Then $\mu_A : X \to [0, 1]$ is a characteristic function of the set $A$ if every $x$

$$\mu_A(x) = \begin{cases} 1 & if\ x \epsilon A \\ 0 & if\ x \epsilon \bar{A} \end{cases} \tag{3}$$

The generalization of the theory of fuzzy sets is: the membership function $\mu_F$ of a fuzzy set $F$ is a function $\mu_F : U \to [0, 1]$

### 2.3.2 Operations with Fuzzy Sets

Notions like equality and inclusion are two fuzzy sets derived from the classic theory of sets. Two fuzzy sets are equal if every element in the universe has the same membership degree in every one of them. A fuzzy set $A$ is a subset of the set $B$ if every element in the universe has a smaller degree of membership.

In the classic theory, the union, intersection and complement are sets of simple operations that are clearly defined. Every logic operator and, or and not has a semantic well defined according to the propositional logic. The most used operators in fuzzy sets are:

$$\mu_A \cap \mu_B(x) = min(\mu_A(x), \mu_B(x)), \tag{4}$$

$$\mu_A \cup \mu_B(x) = max(\mu_A(x), \mu_B(x)), \tag{5}$$

$$\mu_A(x) = 1 - \mu_A(x). \tag{6}$$

This is a simple extension of the classic operations. Other possible extensions are, $\mu_{A\cap B}(x) = \mu_A(x) \cdot \mu_B(x)$ or $\mu_{A\cup B}(x) = min((1, \mu_A(x) + \mu_B(x))$ .

More generally, triangular norms (T-norm and S-norm) are used to represent intersection, union or complement.

### 2.3.3 Fuzzy Relations

A relation can be considered as a set of ordered pairs. As classic sets, classic relations can be described by a membership function. Suppose that $R$ is a relation of $n$ order defined in $X_1 \dots X_n$ then $\mu_R = X_1 \dots X_n \rightarrow [0, 1]$ is a membership function of the set $A$ for every $X_1 \dots X_n$,

$$\mu_R(x_1 \dots x_n) = \begin{cases} 1 & if\ (x_1 \dots x_n) \epsilon R \\ 0 & if\ (x_1 \dots x_n) \epsilon \bar{R} \end{cases} \tag{7}$$

### 2.3.4 Operations with Fuzzy Relations

The two most used operations in fuzzy relations are intersection and union. These are defined as follow: let $R$ and $S$ be binary relations defines in $X \times Y$. The intersection between $R$ and $S$ is defined as

$$\forall (x, y)\, \epsilon\, X\, x\, Y\ :\ \mu_{R\cap S}(x, y) = min(\mu_R(x, y), \mu_S(x, y)). \tag{8}$$

Insted of minimum, any T-norm can be used.

The union of $R$ and $S$ is defined as

$$\forall (x, y)\, \epsilon\, X\, x\, Y\ :\ \mu_{R\cup S}(x, y) = max(\mu_R(x, y), \mu_S(x, y)). \tag{9}$$

Instead of maximum, any S-norm can be used. These definitions can be extended to any number of relations.

The combination of fuzzy sets and fuzzy relations is called composition and is defines as: let $A$ be a fuzzy set defined in $X$ and $R$ a fuzzy relation defined in $X\ x\ Y$. Then, the composition $A$ and $R$ results in a fuzzy set $B$ defined in $Y$ is given by

$$\mu_B(y) = max\ min(\mu_A(x), \mu_R(x, y)). \tag{10}$$

This is the so called **max-min** composition. The **max-prod** composition is defined as

$$\mu_B(y) = max(\mu_A(x), \mu_R(x, y)). \tag{11}$$

### 2.3.5 Approximate Reasoning

The approximate reasoning is the best way, in which, the fuzzy logic covers a variety of inference rules where the premises contain fuzzy propositions.

Inference in approximate reasoning is in contrast to the inference in classic logic. In appoximate reasoning, the consecuence of a set given the fuzzy propositions depends on a esencial way in the aggregate meaning to these fuzzy propositions. Then, inference in appoximate reasoning is the processing of fuzzy sets that represent the meaning of a certain set of fuzzy propositions. For example, given the membership functions $\mu_A$ and $\mu_B$, representing the meaning of a fuzzy proposition $X\,is\,A$ and the meaning of the fuzzy conditional $If\ X\ is\ A\ Then\ Y\ is\ B$, the membership function can be computed representing the meaning of the conclusion $Y\ is\ B$.

### 2.3.6 Inference Rules

In approximate reasoning, two inference rules are the most important. The inference composition rule and the generalized modus ponens. The first rule uses a fuzzy relation to explicitily represent the conection between two fuzzy propositions, the second uses a rule IF-THEN that implicitly represents a fuzzy relation. The generalized modus ponens has the inference symbolic scheme

$$S_1\ is\ Q_1;\ if\ S_1\ is\ P_1\ then\ S_1\ is\ P_2\ S_2\ is\ Q_2$$

where $S_1$ and $S_2$ are symbolic names for linguistic variables, and $P_1$, $P_2$, $Q_1$ and $Q_2$ are symbolic names for linguistic values. The rule of composition for inference can be considered as a special case of generalized modus ponens. The general symbolic form is

$$S_1\ is\ Q_1\ S_1\ R\ S_2\ S_2\ is\ Q_2$$

where $S_1\,R\,S_2$ is read as "$S_1$ sin in relation $R$ to $S_2$" and the meaning is represented as a fuzzy relation. Then, instead of the IF-THEN rule, there is a fuzzy relation $R$. Now the scheme of inference is considered,

$$S_1\ is\ P\ S_1\ R\ S_2\ S_2\ is\ Q$$

where $P$ and $Q$ are fuzzy sets representing the meaning of $P$, $Q$ and $R$ are a fuzzy relation defining the meaning of $R$, $P$ is defined in $X$ and $R$ over $X\ \times\ Y$. Then the calculation of the compositional rule of inference is done as

$$\mu_Q(y) = max_x min(\mu_P(x), \mu_R(x,y)). \tag{12}$$

### 2.3.7 The IF-THEN Rules

There are a number of relations that can represent the meaning of IF $X$ is $A$ THEN $Y$ is $B$. The most used implications in fuzzy sets are:

Implication of Lukasiewicz: This implication is based on the equalivalence $p \rightarrow q \equiv' p \vee q$. To represent OR is also possible to use the limited sum $min(1, 1-p+q)$ instead of the maximum $max(p,q)$. This results in the relation called $R_a$, defined as

$$\mu_{R_a}(x,y) = min(1, 1 - \mu_A(x) + \mu_B(y)). \tag{13}$$

Implication of Zadeh: In logic of two values, $p \rightarrow q$ has the same true values as $(p \Lambda q) \vee' p$. This equivalence was used by Zadeh in the next form

$$\mu_{R_m}(x,y) = max(min(\mu_A(x)), \mu_B(y)), 1 - \mu_A(x)). \tag{14}$$

Implication of Mamdani: With respect to fuzzy control, this is the most important implication known in the literature. Its definition is based in the intersection. The relation $R_c$ (x of conjuction) is defined as

$$\mu_{R_C}(x,y) = min(\mu_A(x), \mu_B(y)). \tag{15}$$

### 2.3.8 Fuzzification (FM)

According to [24], the fuzzyfication module does the next functions:

- FM-F1: Does a scale transformation (input normalization) that maps the physical values of the input variables in a normalized universe (normalized domain). It also maps the normalized value of the output variable in the physical domain. When a normalized domain is used, it is no necessary to use FM-F1.
- FM-F2: Does the defuzzification that converts the recent value of an input variable of a fuzzy set to make it compatible with the representation of the fuzzy set of the input variable.

### 2.3.9 Inference Machine

There are two ways to attack the design of the inference machine of a fuzzy controller: (1) inference based in composition and (2) inference based on individual rules. The basic form of the inference machine of the second type is to compute the general value of the output variable in individual contributions for each rule in the base of rules. Each individual contribution represents the values of the computed output variables for each individual rule. The output of the fuzzification module, represents the actual values of the input variables and are projected to each rule, and a certain degree of equivalence is stablished. Each degree of equivalence represents the degree of satisfaction of a fuzzy proposition. Based on the degree of equivalence, the value of the output variable in the last rule is modified. The set of all the output values of the equivalent rules represents the general value of the fuzzy output. In this context, the design of parameters for the inference machine is:

- Choose the representation of the meaning for a single rule,
- Choose the representation of the meaning for a set of rules,
- Choose the inference machine,
- Prove the set of rules to be consistent and complete.

### 2.3.10 Defuzzification (DM)

According to [24], the functions of the defuzzification module are:

- DM-F1: Does the so called defuzzification that converts the set of modified output values to a single value.
- DM-F2: Does the denormalization of the output that maps the output points in the physical domain. DM-F2 is not necessary if domains not normalized are used.

Two of the most used defuzzification operators are Center of Area and Mean of Maximum.

Center of Area: The method of center of area or gravity center is the most known defuzzification method. It is almost discrete, this results in

$$y^* = \frac{\sum_{i=1}^{l} y_i \cdot \mu_Y(y_i)}{\sum_{i=1}^{l} \mu_Y(y_i)} = \frac{\sum_{i=1}^{l} y_i \cdot max_k \mu_{CLY(k)}(y_i)}{\sum_{i=1}^{l} y_i \cdot max_k \mu_{CLY(k)}(y_i)} \tag{16}$$

In the continues case, it is obtained

$$u^* = \frac{\int_y u \cdot \mu_Y(y) dy}{\int_y \mu_Y(y) dy} = \frac{\int_y y \cdot max_k \mu_{CLY(k)}(y) dy}{\int_y max_k \mu_{CLY(k)}(y) dy} \tag{17}$$

Mean of Maximum: This method determines the first and last values where $Y$ has a maximum degree of membership and then it takes the mean of those two values. Formally,

$$y* = \frac{\{inf\ y \epsilon \gamma \mid \mu_Y(y) = hgt(Y)\} + sup\{inf\ y \epsilon \gamma \mid \mu_Y(y) = hgt(Y)\}}{2} \tag{18}$$

## 2.4 Real-Time Systems

Dealing with computer image processing is a challenging task in vehicle navigation since high number of computations need to be done in order to process the information from the cameras and still have time to control the vehicle around the desired trajectory.

Real-time computing systems are systems in which the importance of an action is not only that it is done correct, but also the time it takes to be processed. In order for tasks to get done at exactly the right time, real-time systems must allow you to predict and control when tasks occur [1].

A real-time system must demonstrate the following features:

- Predictably fast response to urgent events.
- High degree of schedulability: the timing requirements of the system must be satisfied at high degrees of resource usage.
- Stability under transient overload: when the system is overloaded by events and it is impossible to meet all the deadlines, the deadlines of selected critical tasks must still be guaranteed.

### 2.4.1 Real-Time Scheduling Policies

There are different schemes for scheduling events. According to [1], some popular real-time scheduling policies include:

- Fixed Priority Preempitve Scheduling: Every task has a fixed priority that does not change unless the application specifically changes it. A higher-priority task preempts a lower-priority task. Most real-time operating systems support this scheme.

- Dynamic-Priority Preemptive Scheduling: The priority of a task can change from instance to instance or within the execution of an instance, in order to meet a specific response time objective. A higher-priority taks preempts a lower-priority task. Very few commercial real-time operating systems support such policies.

- Rate-Monotonic Scheduling: An optimal fixed-priority preemptive scheduling policy in which, the higher the frequency of a periodic task, the higher its priority. This policy assumes that the deadline of a periodic task is the same as its period. It can be implemented in any operating system supporting fixed-priority preemptive scheduling or generalized to aperiodic tasks.

- Deadline-Monotonic Scheduling: A generalization of the rate-monotonic scheduling policy in which the deadline of a task is a fixed point in time relative to beginning of a period. The shorter this (fixed) deadline, the higher its priority. When the deadline time equals the period, this policy is identical to the rate-monotonic scheduling policy.

- Earliest-Deadline-First Scheduling: A dynamic-priority preemptive scheduling policy. The deadline of a task instance is the absolute point in time by which the instance must complete. The deadline is computed when the instance must complete. The scheduler picks the task with the earliest deadline to run first. A task with an earlier deadline preempts a task with a later deadline. This policy minimizes the maximum lateness of anyset of tasks relative to all other scheduling policies.

- Least Slack Scheduling: A dynamic-priority non-preemptive scheduling policy. The slack of a task instance is its absolute deadline minus the remaining worst-case execution time for the task instance to complete. The scheduler picks the task with the shortest slack to run first. This policy maximizes the minimum lateness of any set of tasks.

# 3 State of Art

In this chapter, related works are presented in order to stablish the context on which this thesis was based. The main areas of study of this thesis are: road detection using stereo vision and fuzzy control in autonomous mobile vehicles.

## 3.1 Road Detection with Stereo Vision

Vision-based trajectory detection is a very important area of study because it is a fundamental part in autonomous driving, car collision warning, object detection and pedestrian crossing detection. Detecting trajectories and roads with vision systems can be done using monocular vision-systems and stereo vision systems [15, 3]. Particularly, stereo vision has been studied lately because it gives an advantage over monocular vision systems: the measurement of distance, without the need of more sensors; therefore, we focus on the research of stereo vision to detect the trajectory where the vehicle is driven. Great interest has recently arisen in the design and development of autonomous land vehicle [34, 35]. Two of functions of ALV(Autonomous Land Vehicle) autonomous navigation are the obstacle detection and the robust detection and tracking of road boundaries.

Road detection is a crucial problem for intelligent vehicles and mobile robots. It provides information about the world that enables the intelligent vehicle or robot to interact with its enviroment and react to events or changes that influence its task [44]. Many researchers have been studying it for several decades and dramatic development has been accomplished, which can be categorized into two main types of methods: vision-based methods and LIDAR(Light Detection And Ranging)-based methods. The stereovision makes possible to use only cameras to directly measure range and color information, just like human operators. Therefore, vision-based road detection is a very important as well as promising branch in the field [15]. Among the curren vision-based methods, some use monocular camera to extract the road region by employing features with specific intensity, color and texture while others use a binocular camera for road detection by using 3D structural information [15].

A variety of methods have been proposed for obstacle detection. Several kinds of sensors are used to acquire information from the environment to carry out robot navigation with real-time obstacle avoidance. Vision system, 2D or 3D laser rangefinder and combinations of them are used to detect obstacles under different environment. Stereo vision technique [43] was popularly used to detect obstacles for ALV. The main problem of stereo vision is that complex algorithm has to be used to guarantee the correct pixel matching between two images. In the past several years, the laser range measurement system has been used to detect obstacle. In [58], an obstacle detection system for ALV under semi-structural environment with two 2D laser range finders is described. 3D LRF(Laser Range Finder)[38] and quasi-3D LRF [26] is used for obstacle detection in cross-country environment and urban environment.

Autonomous road following can be considered as five different parts: horizon estimation, obstacle detection, obstacle avoidance, road detection and 3D view as shown

in figure 4.

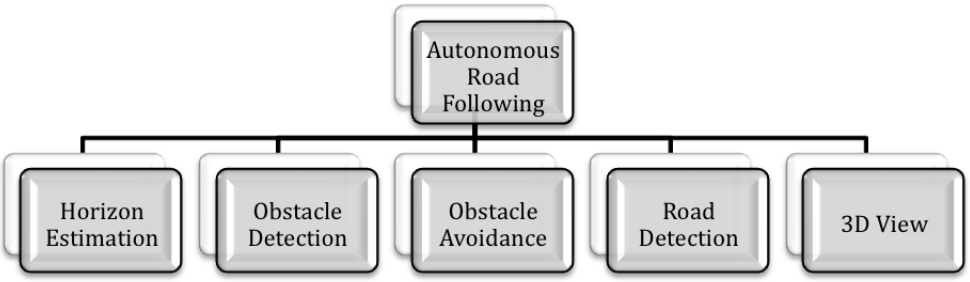


Figure 4: Components of autonomous road following

### 3.1.1 Horizon Estimation

The horizon line is important information for knowing the area of interest in the image. The road will be usually below the horizon line. To estimate the position of the horizon line, an approach has been introduced by [32, 28, 52]. This method estimates the horizon line by applying non-linear mixtures of linear regressors to the description of an image obtained using gist descriptors [4]. Also, the horizon line can be easily detected by detecting the road lines and extend them to know the point where those two lines intersect; above that point, there is no information about the road, and below that point is the area of interest. A vanishing point is computed at that intersection to differentiate the road from the horizon gradually [3]. Figure 5 shows the horizon line estimation according to [3].


Figure 5: Detection of the horizon line according to [3].

Detecting the road using the vanishing point is robust to global lightning variations, different road types, damaged roads and the presence of other vehicles in the scene. However, it is not robust against curved roads, heavy traffic and when strong shadow edges are present [3].

### 3.1.2 Obstacle Detection

When dealing with obstacles, there are a number of ways to detect them. One method is to use road-barriers [37], another method is to use uncertainty [53] and the principle of color declivity [11].

According to [24], the obstacles can be classified as positive and negative. Positive obstacles are tree trunks, sand dunes and others that extend out of the ground surface. The other type are the negative obstacles, such as ditches or holes, that extend into the ground plane. Positive obstacles are detected by applying a hysteresis threshold on the measured terrain slope around image points according to [24]. Negative obstacles are detected by looking for depth jumps in the range profile of an image column. In order to follow roadside under various condition, vision-based sensor and range-based sensors are used. In [44], an obstacle detection method integrates information from laser rangefinder and camera to detect and track obstacles. In[3], a road following method integrates information from laser rangefinder and camera to detect and track the road boundary. Road height, smoothness, color, and texture were combined to yield higher performance of roadside.

### 3.1.3 3D View

Another 3D cue is the layout of the scene. The layout is analyzed using three major parts of the image: (1) sky pixels, (2) vertical surface pixels and (3) ground pixels. With these 3D cues the road is limited to ground, non-sky image regions. Further, regions are avoided which are vertically orientated (i.e., buildings, vehicles, pedestrians or any other object present in the scene). The segmentation of the image in these 3D cues is computed by the method proposed in [16]. Road detection using scene layout is robust to different types of asphalts, lane markings and pedestrian crossings. However, scene-layout for road detection may be sensitive to shadows as the algorithm uses superpixel segmentation [3]. Another important cue for detecting the road is its 3D geometry. This road geometry can be inferred using a scene (road) classification algorithm where each class represents typical 3D road geometries such as left turn, straight road and junctions [30].

## 3.2 Fuzzy Control in Autonomous Mobile Vehicles

Control algorithms should be considered as an important issue in road following to ensure safe and smooth rides. Although a lot of researches have been done, most of them are based on traditional control theory such as PID [56] and linear controllers [6]. The kinematic behavior of autonomous road following is typically nonlinear. Therefore linear models usually fail to describe these systems efficiently. However, it is difficult to analyze nonlinear mathematical models for autonomous road following schemes. Other methods such as neural-networks [5, 17] and reinforcement learning (RL) [48] approaches have also been used in road following, but these approaches require learning which consumes extra computation time. Human drivers can drive a car smoothly with their driving expertise rather than knowledge about control theory. Fuzzy logic

control is known to be an organized method to emulate human expertise in dealing with imprecise data. It attempts to apply a human-like way of thinking in the application areas and allows linguistic terms for intermediate values to be defined besides conventional evaluations. Fuzzy logic control has been[10] applied in autonomous road following by some researchers [51, 31, 10]. In [59] a method is proposed to optimize the road following fuzzy controller.

## 3.3 Comparison of Works

A comparative of the works in the area has been realized in order to put the work done in context. Table 1 shows this comparative in the last 6 years. The parameters of differenciation are, the type of vision used, if the work includes detection and/or avoidance of obstacles, the type of control if used and if there was an implementation of the system.

Table 1: Comparison of the different works in recent years.

| Year of Publication | Autor(s) | Title | Vision | Obstacle | Control | Implementation |
|---|---|---|---|---|---|---|
| 2005 [19] | Broggi, A.; Caraffi, C.; Fedriga, R.I.; Grisleri, P. | Obstacle Detection with Stereo Vision for Off-Road Vehicle Navigation | Stereo | Detection | None | Yes |
| 2006 [45] | Seung-Hun Kim; Chi-Won Roh ; Sung-Chul Kang ; Min-Yong Park ; | A Hybrid Autonomous / Teleoperated Strategy for Reliable Mobile Robot Outdoor Navigation | Mono | | | |
| **2006 [24]** | Cabani, I.; Toulminet, G.; Bensrhair, A. | A Fast and Self-adaptive Color Stereo Vision Matching | Stereo | None | None | No |
| 2006 [43] | Zezhong Xu; Yanbin Zhuang; Huahua Chen ; | Obstacle Detection and Road Following using Laser Scanner | Lasser | None | None | Yes |
| 2006 [44] | Perrollaz, M. ; Labayrade, R. ; Royere, C. ; Hautiere, N. ; Aubert, D. ; | Long Range Obstacle Detection Using Laser Scanner and Stereovision | Laser/ Stereo | Detection | None | Yes |
| **2007 [22]** | Dubbelman | Obstacle Detection during Day and Night Conditions using Stereo Vision | Mono | Detection | None | Yes |
| **2007 [20]** | van der Mark, W. ; van den Heuvel, J.C. ; Groen, F.C.A. ; | Stereo based Obstacle Detection with Uncertainty in Rough Terrain | Stereo | Detection | None | Yes |
| 2008 [50] | Hong, D. ; Kimmel, S. ; Boehling, R. ; Camoriano, N. ; Cardwell, W. ; Jannaman, G. ; Purcell, A. ; Ross, D. ; Russel, E. ; | Development of a semi-autonomous vehicle operable by the visually-impaired | Stereo/ Laser/GPS | None | None | Yes |
| 2008 [48] | Neagoe, V. ; Tudoran, C. ; | Road following for autonomous vehicle navigation using a concurrent neural classifier | Mono | None | Neural Classifier. | Yes |
| 2009 [21] | Tiberiu Marita | Barriers Detection Method for Stereovision-Based ACC Systems | Stereo | Detection | None | Yes |
| 2009 [46] | Lidoris, G. ; Rohrmuller, F. ; Wollherr, D. ; Buss, M. ; | The Autonomous City Explorer (ACE) project — mobile robot navigation in highly populated urban environments | Laser | Detection/ Avoidance | Behavior Selection. | Yes |
| 2009 [9] | Yi Fu; Li, H.; Kaye, M. | Design and Stability Analysis of A Fuzzy Controller for Autonomous Road Following | Mono | None | Fuzzy | Yes |
| **2010 [49]** | Das, A.; Naroditsky, O. ; Zhiwei Zhu ; Samarasekera, S. ; Kumar, R. ; | Robust visual path following for heterogeneous mobile platforms | Stereo | None | Follower | Yes |
| 2010 [47] | Yi Fu ; Li, H. ; Kaye, M.E. ; | Hardware/Software Codesign for a Fuzzy Autonomous Road-Following System | Mono | None | Fuzzy | Yes |
| 2010 | Hernandez, Aristeo. | Stereovision Feedback and Fuzzy Control for Autonomous Robot Navigation. | Stereo | Detection/ Avoidance | Fuzzy | Yes |

It can be concluded that there has been a lot of research in stereovision in recent years, although this work of thesis uses stereovision, it also includes detection and avoidance of obstacles, which differentiates it from most of previous work. The type of control to be used is in fuzzy logic, which not many works select this and the work is implemented in order to be tested and make experiments. As far as we know according to the comparison of the work, it is different to what it has been done in recent years and also the techniques used in stereo vision are different from the classical ones, and we think, it is a major contribution.

# 4 Proposal

In this chapter, the proposal for solution is presented and the algorithm that was used in order to detect the road, as well as the method for controling the car's direction with the fuzzy controller and the obstacle detection to stop the vehicle. The reasons for using each method are explained and some of them, compared to others in order to show better performance. The solution to solve the problem of this thesis is presented in this chapter. Figure 6 presents the individual components to solve the problem.

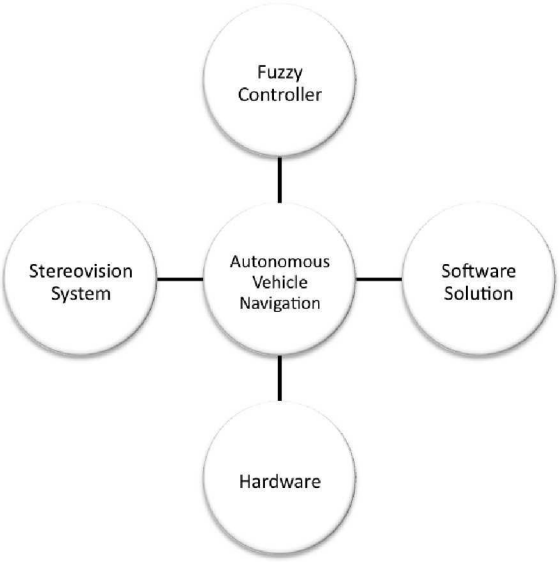


Figure 6: Integral solution

## 4.1 Stereovision system

This part of the project is fundamental, because the stereocamera is the sensor of the robot and the eficiency of the total system depends on the recognition of the road and the image processing; the output of this process is the input of the fuzzy controller. The stereovision system consists of two webcam cameras that are aligned to be considered as a stereocamera. A stereo camera grabs video of a certain scene but theres an offset between the two images because of the different placement of each camera. This offset is calibrated and then correlated between the two cameras to know the distance of an object in the image.

### 4.1.1 Image acquisition

The image is acquired by two cameras placed in front of the car. Each camera acquires the segment of the road in front of the robot, but with an offset between them. The grab acquisition is calibrated so that each frame in each camera is taken at the same time to process the image correctly. The interface with the cameras is through USB.

### 4.1.2 Image processing

After acquiring the two images, they need to be processed to get the necessary information to recognize the road. First, a threshold is applied to the original image in order to eliminate noise and emphasize the road. Then, brightness, contrast and gamma are equalized according to the scene to have a better recognition of the road lines. Having processed the image, the edge detection is done, three lines are constantly analyzed to detect the edges. With the information of the edges detected, the points a, b and c are calculated, which are the parameters to calculate $\theta$ and $\rho$. Figure 7 shows this procedure.


Figure 7: Flowchart of the image processing.

An algorithm developed was implemented to detect the road. Basically, there are four horizontal lines that are constantly checked in each image to determine the edges. The far top line is to detect objects in the long distance in order to react against them properly because this line is the one that detects first the objects that pass trough it. The other three lines are used to detect the road itself. They are called Top Line, Center Line and Bottom Line, and the reason to use three lines is because they are the fewest necessary to know the angle $\vartheta$, which determines the angle of curvature of the road, and fewer image processing means faster computer processing. When detecting the edges, 4 points are detected for each line and the objective is to get the points a, b and c which are in the medium of the line segments. This points are calculated as follows,

$$a_x = \left[\frac{p_x1 + p_x2 + p_x3 + p_x4}{4}\right] \tag{19}$$

$$b_x = \left[\frac{p_x5 + p_x6 + p_x7 + p_x8}{4}\right] \tag{20}$$

$$c_x = \left[\frac{p_x9 + p_x10 + p_x11 + p_x12}{4}\right] \tag{21}$$

$$a_y = C_1 \tag{22}$$

$$b_y = C_2 \tag{23}$$

$$c_y = C_3 \tag{24}$$

The points $a_y$, $b_y$ and $c_y$ are constant, because they are fixed coordinates that are always checked. Points $a_x$, $b_x$ and $c_x$ that represent the center point of the road, are used to calculate the slopes between a-b, and b-c. The slopes are converted to angles where a vertical line represents $0^\circ$. The difference in angles between a-b and b-c shows the curvature of the road. Slopes are calculated as follows,

$$S_{a-b} = -\frac{b_y - a_y}{b_x - a_x} \tag{25}$$

$$S_{b-c} = -\frac{c_y - b_y}{c_x - b_x} \tag{26}$$

And to know the angle $\theta_R$, which determines the angle of curvature of the road,

$$\theta_R = (atan(S_{a-b}) - atan(S_{b-c}))\left(\frac{180}{\pi}\right) \tag{27}$$

The setpoint will always be $\theta_R$. The angle of the vehicle with respect to the road is $\theta_v$ which is determined between the angle of the line b-c and the vertical line. It can be positive or negative, according to the orientation of the vehicle with respect to the road.

$\rho$ is calculated as the point $c_x$. This way, it is a relative distance instead of a fixed distance to the origin of the image because the positioning depends on the orientation of the vehicle according to the road at that point. The setpoint for $\rho$ is always 320, which is the center of the x-axis because of the resolution of the webcam (480x640). Figure 8 shows the points detected and calculated with this method.

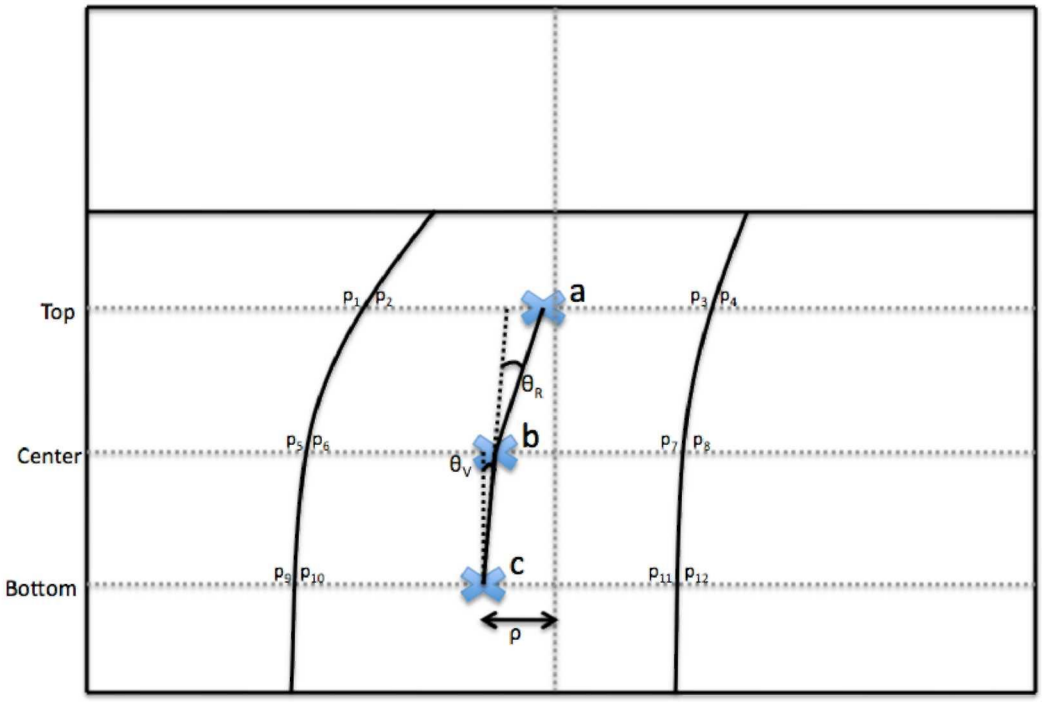


Figure 8: Points detected and calculated in the image processing

When using a stereo camera, the parameters for both frames are calculated as the average value of each parameter.

## 4.2 Fuzzy Controller

The fuzzy controller is the brain of the robot and the processing done determines the manipulation done to the actuators in order to keep the vehicle in the desired position. The inputs of the controller are $\vartheta$ and $\varrho$, which are calculated as shown in section 4.1 after the image processing and road recognition.

In the present work, a fuzzy controller was developed to keep the car inside the road and four linguistic variables were used: the angle of the car relative to the angle of the road $\theta$ and the distance from the car's center to the road center line $\rho$. The linguistic expressions used to describe the magnitude of the linguistic variables contain the following basic adjectives: negative very big (-VB), negative big (-B), negative small (-S), zero (o), positive small (+S), positive big (+B) and positive very big (+VB). This means that the set of every linguistic variable contains as many elements as the number of adjectives used to describe the variable. The cardinality of this set denotes the number of elements in this set. These sets are the basis for the fuzzification and defuzzification procedures. Since no expert knowledge was available, the rule base proposed by [36] was implemented using the above basic adjectives. According to [23], the Mandani implication is the most important implication (IF-THEN rule) known in fuzzy control literature. This fact explains why the Mandani implication was employed

in the present development.

According to [23], the control rules are designed based on expert knowledge and testing. Furthermore, the control rules also meet the stability requirements derived from Lyapunov's direct method. For example, if $\rho$ is +B and is increasing rapidly $\theta$, then the vehicle should turn left, i.e. $\theta_S$ should be +B. Based on this knowledge, we can obtain twenty five fuzzy rules. Table 2 represents abstract knowledge that an expert uses to control the steering angle given from information about the error of $\theta$ and $\rho$ . The input and output linguistic variables are summarized in the table.

Table 2: Set of rules of the fuzzy controller.

| θ \ ρ | ++ | + | 0 | - | -- |
|---|---|---|---|---|---|
| ++ | -VB | -VB | -B | +S | +S |
| + | -VB | -B | -S | +S | +B |
| 0 | -B | -S | 0 | +S | +B |
| - | -B | -S | +S | +B | +VB |
| -- | -S | -S | +B | +VB | +VB |

Note: VB: Very Big; B: Big; S: Small; 0: Zero.

In order to apply the fuzzy control, $\theta$ and $\rho$ need to be determined; these are calculated in the image processing part and are the input for the fuzzy control. Fuzzification need to be done in order to map the inputs from crisp values to grades of membership for linguistic terms of fuzzy sets and then according to the rule base, a degree of truth is calculated; the defuzzification process maps the values to real values for the manipulation. The process for the fuzzy controller is shown in figure 9.

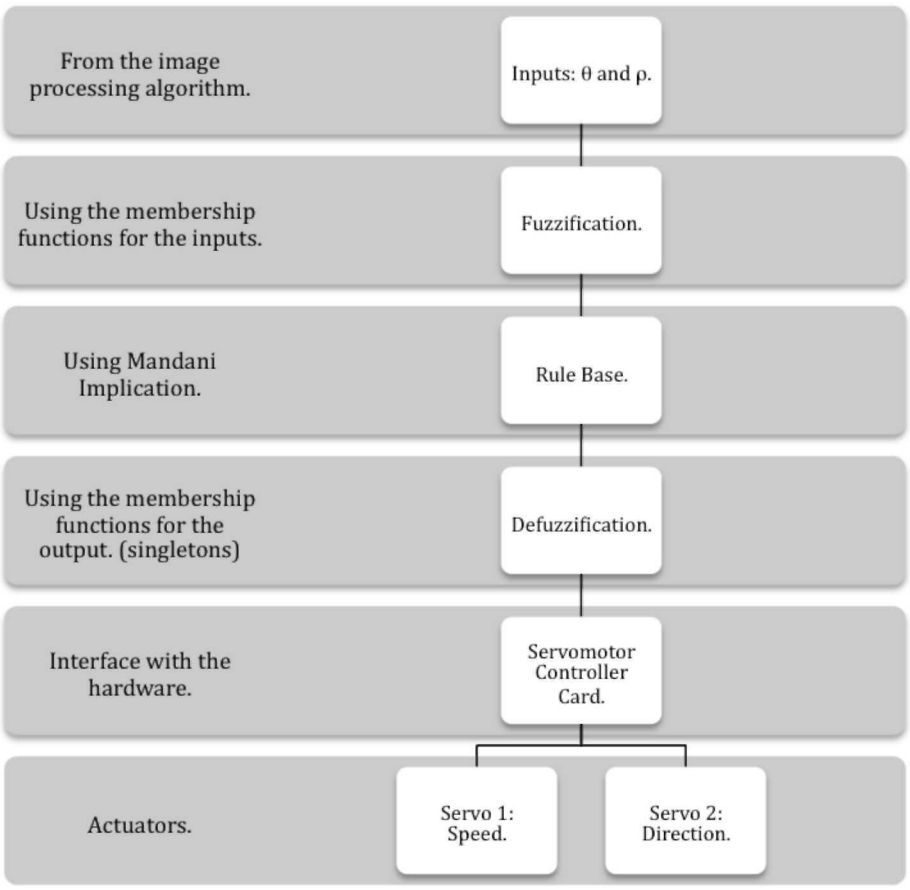


Figure 9: Flowchart of the fuzzy controller.

A block diagram of the fuzzy controller is shown in Fig. 10. The desired orientation of the center line of the car should be aligned with the road centroid. The error is the angle between the desired orientation of the center line and the actual center line of the car. To reduce the error to zero, the steering angle should be equal to the angle of the road. The fuzzy controller is done in a program in LabVIEW which it first does the fuzzification to the inputs, and with the rule base in Table 2 the inference mechanism is applied. The defuzzification is applied to those results from the inference mechanism and it is sent to the actuators of the vehicle to keep it in the desired position.

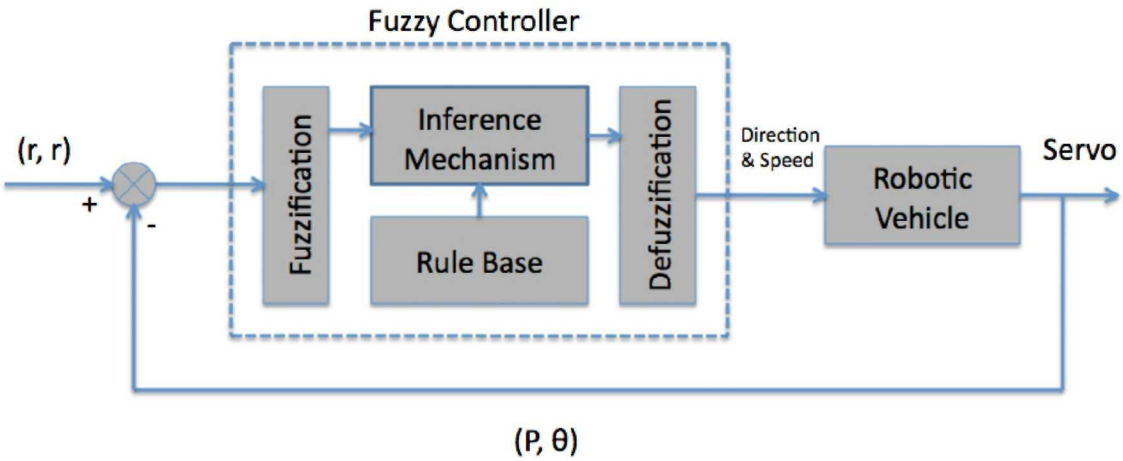


Figure 10: Block diagram of the fuzzy logic controller.

The fuzzification procedure maps the crisp input values to the linguistic fuzzy terms with the membership values between 0 and 1. The fuzzification method used is max-min. In this thesis, we use five membership functions for both error in $\rho$ and error in $\theta$. Figures 11 and 12 illustrate the normalized input membership functions for $\rho$ and $\theta$ respectively.

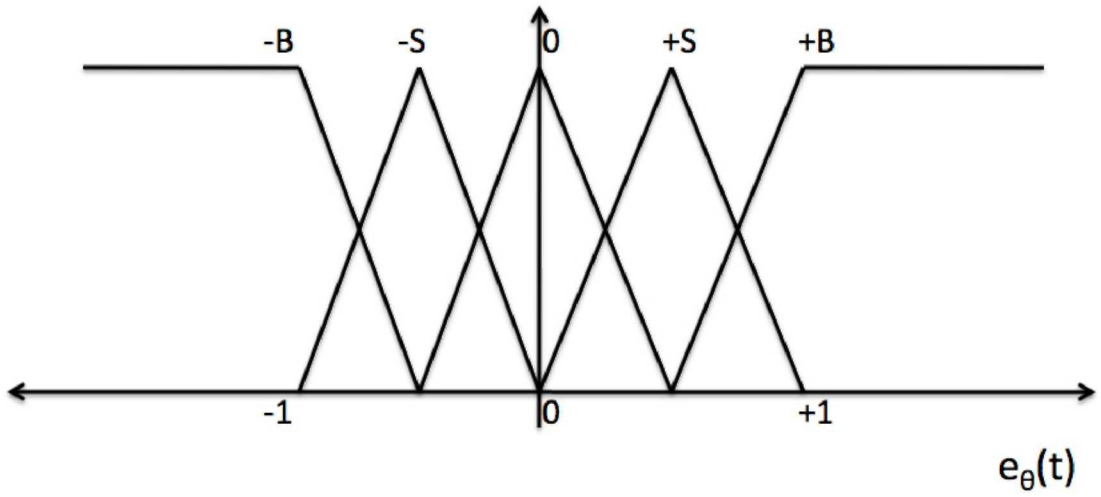


Figure 11: Input membership functions for error in $\rho$.

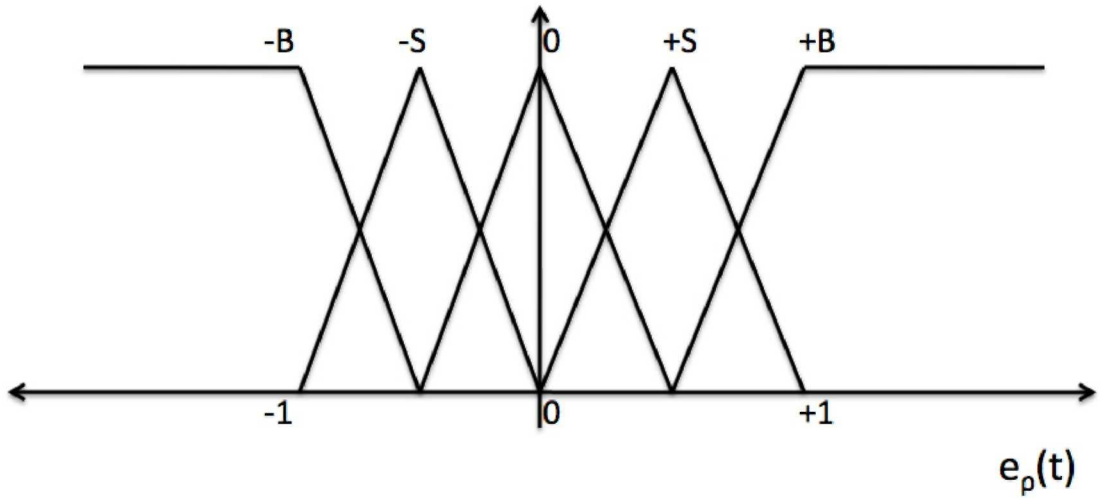


Figure 12: Input membership functions for error in $\theta$.

The defuzzification procedure maps the fuzzy output from the inference mechanism to a crisp signal. It is used Center of Sums as the defuzzification method to combine the recommendations represented by the implied fuzzy sets from all the rules. Fig. 13 shows the normalized output membership functions. In this case, for faster computational speed, the type of membership functions are called singletons, each one has the value of 1.

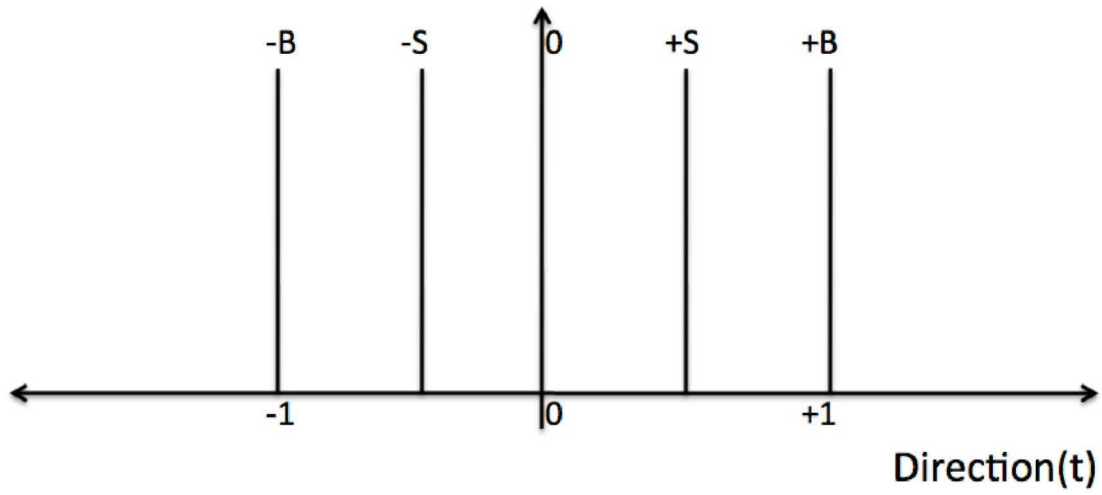


Figure 13: Output membership functions.

## 4.3 Software Plataform

In order to develop the code to integrate the stereovision system and the fuzzy controller with the hardware, the use of the program LabVIEW is proposed mainly because its hardware integration and the ability to grab images from two cameras of the same model through USB. The data display, custom controls and the user interface are also aspects analyzed to choose this plataform for designing. Appendix B shows more about this software platform and the actual code to implement this proposal.

## 4.4 Hardware

To implement this thesis and to carry out the experiments, the structure of an R/C car is proposed. It has de basic components desired: an steering system and a motor for the traction. The other elements of the car are not used; instead, elements would be added to complete the objective. This components would be explained in chapter 5, and the specifications of the hardware is shown in Appendix A.

# 5 Experiments and Results

In this chapter, all the experiments done in order to prove the proposal are presented. Also, the components used, and the infraestructure of the project are shown. In table 3, a list of the experiments done and presented in this section is explained.

Table 3: Experiments carried out

| Experiment | Objective |
|---|---|
| Road Detection | To recognize the road accurately in different situations. |
| Distance Calculation | To calculate the distance between the camera and the different points accurately. |
| Object Detection | To detect objects present on the surface of the road. |
| Controllability | To know that the car maintains its position across the road. |

## 5.1 Hardware

The robot was built using a remote control car as a base, but the structure was modified in order to leave space for the actuators and sensors. Basically, only the steering part, and the wheels of the remote control car were used. The basic structure is shown in figure 14.

Figure 14: Basic structure of the mobile robot.

### 5.1.1 Actuators

The main actuators of the mobile robot are two servomotors and one DC motor. One servomotor is attached to the steering system, in order to modify the steering angle according to the fuzzy controller output. In figure 15 the servomotor is shown with the steering system.

Figure 15: Servomotor attached to the steering system.

The other servomotor is used to modify the speed of the car. This servomotor is controlled to move a certain angle; the servomotor is coupled to a potentiometer that regulates the input voltage of the microcontroller, which converts it into a PWM signal that regulates the voltage of the DC motor in order to speed it up or down. Figure 16 shows the schematic diagram of this configuration.

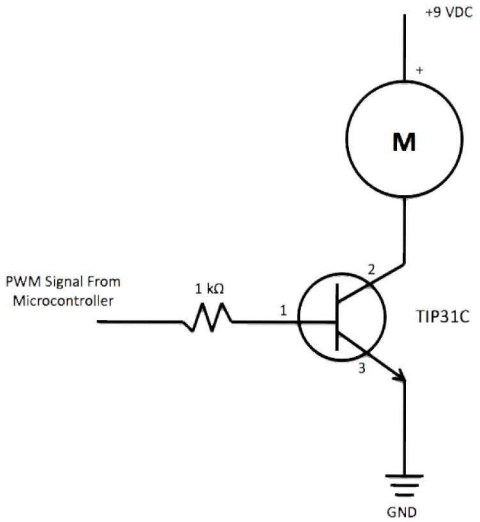


Figure 16: System that regulates the speed of the robot.

The servomotors are controlled using a Servomotor Controller Card made by Lynx-

motion. This controller card is able to manipulate 32 different servomotors, although in this case, we use only two. The card communicates to a computer through a serial cable. Figure 17 shows the schematic diagram of the servomotor controller card with the servomotors.

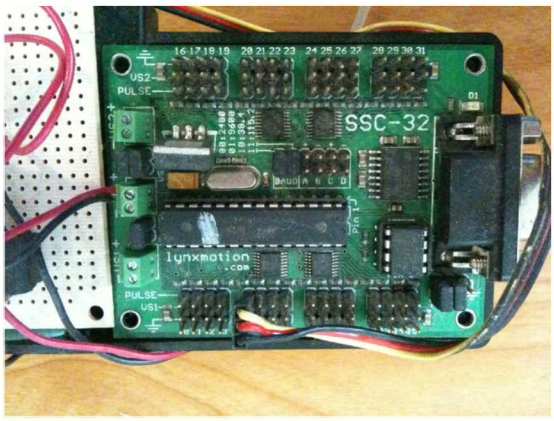


Figure 17: Servomotor controller card configuration.

In order to control the speed of the car, a process needs to be done from the manipulation signal to the actuator, which is a DC motor mounted on the rear wheels. From the fuzzy control process, a manipulation signal is sent to the servo controller, in order to control servo 1, that moves proportionally to the desired speed. This change in position is detected with a potentiometer that sends a regulated voltage between 0 and 5 Volts to the microcontroller. The microcontroller converts that voltage into a PWM signal that is applied to a transistor TIP31C. According to the width of the pulse, the transistor is turned on and off with a certain frequency; according to the frequency, the DC motor is provided with a voltage between 0 and 9 VDC that makes the car move faster or slower. Figure 18 shows the block diagram of this process.

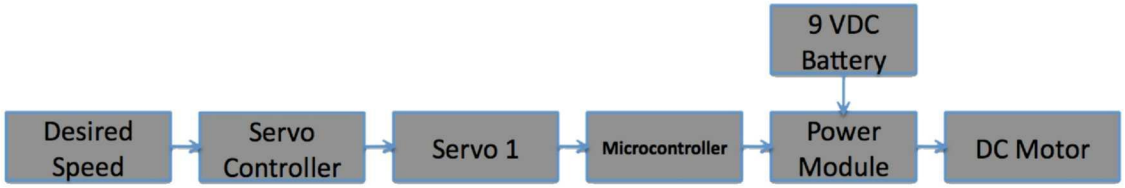


Figure 18: Block Diagram of the Speed Control.

The steering system works something like the speed control. First, the desired steering angle is calculated in the fuzzy controller. The manipulation needed to the direction of the car is sent to the servo controller that converts the signal in order to move the servo 2 as desired. Because the servo 2 is attached directly to the steering system, the wheels are moved proportionally. Figure 19 shows this procedure.

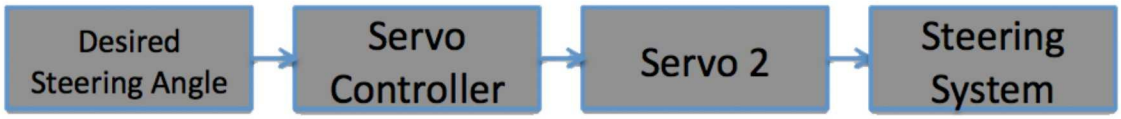


Figure 19: Block Diagram of the Steering System.

### 5.1.2 Sensors

The autonomous mobile robot has two sensors. These are two webcams integrated in front of the car and positioned parallel to each other to grab a video sequence to make the stereovision analysis. The webcams are two Microsoft LifeCamVX-2000 that are able to grab at 30 fps with a resolution of 640 x 480 pixels. Figure 20 shows the two sensors of the car.

Figure 20: Stereocamera.

### 5.1.3 Processing

For the image processing, and the calculation of the fuzzy controll, a mini laptop was used. The first approach for this part, was to use rather a Motherboard or an Embedded Computer, but these two were more expensive than a netbook and offered less specifications for what was needes. The computer was mounted on top of the car while the experiments were made. The computer used was a Dell Inspiron 300m with a processor Centrino at 1.6 Ghz and 1 Gb of RAM.

## 5.2 Image Processing and Road Detection

The algorithm described in section 4.1 was used to process the image grabbed from the stereocamera and detect the road.

In figures 21-24, the results of the image processing are presented. Each image contains the left and right frames and the image of the road. The small yellow points represent the edge detection in each of the three lines and the yellow lines represent the connections between a-b and b-c. Between the two frames, the Theta and Rho of the system are calculated and the distance from the stereocamera to the points a, b and c is shown.

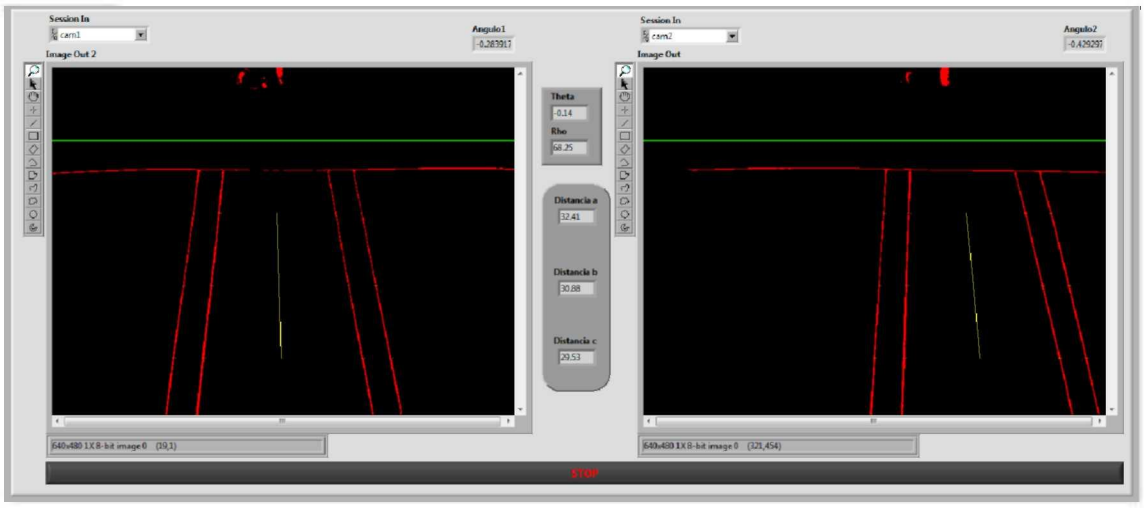


Figure 21: Front Panel of the vehicle on a straight line with positive Rho.

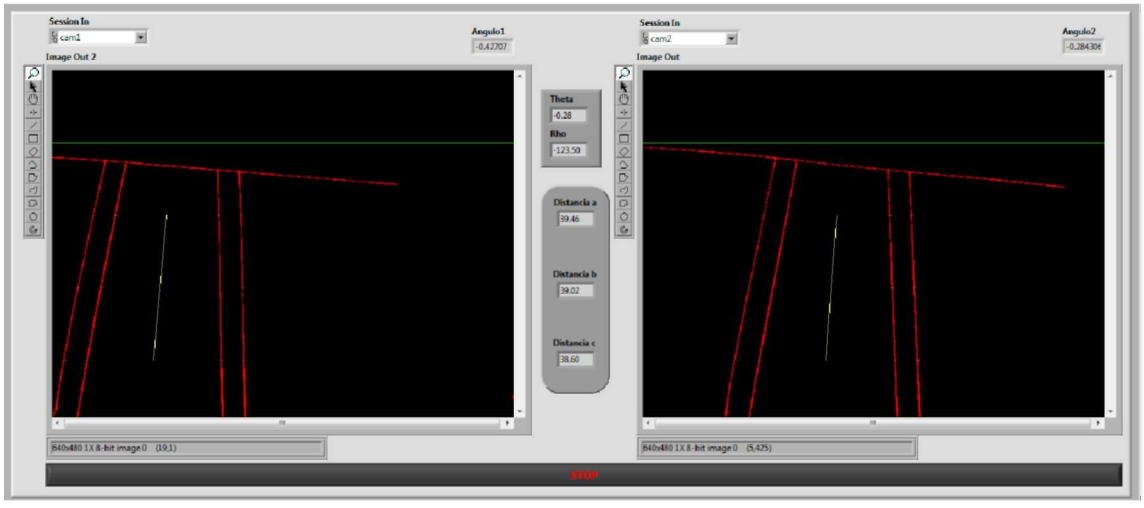


Figure 22: Front Panel of the vehicle on a straight line with negative Rho.

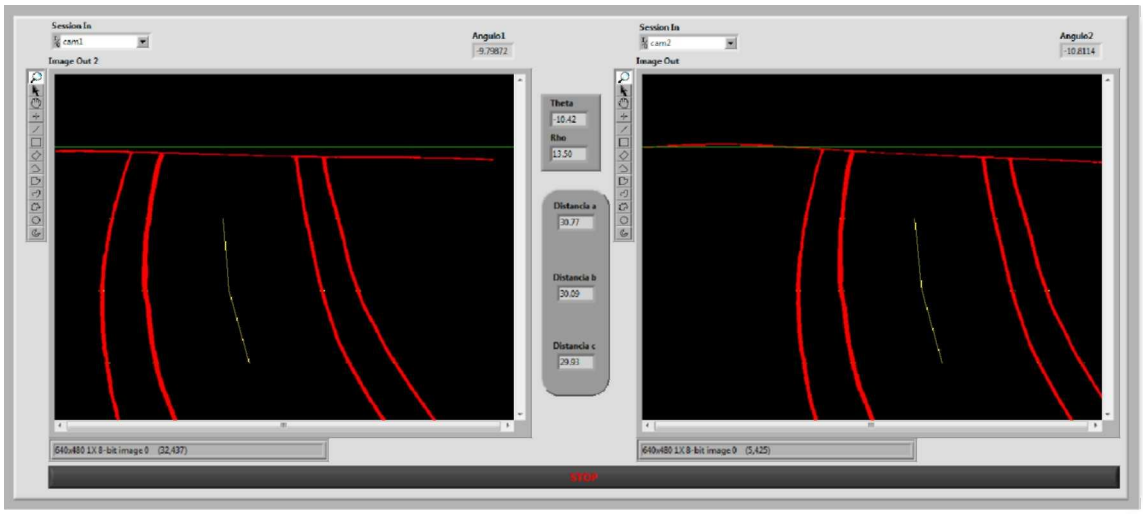


Figure 23: Front Panel of the vehicle on a curve with negative Theta.

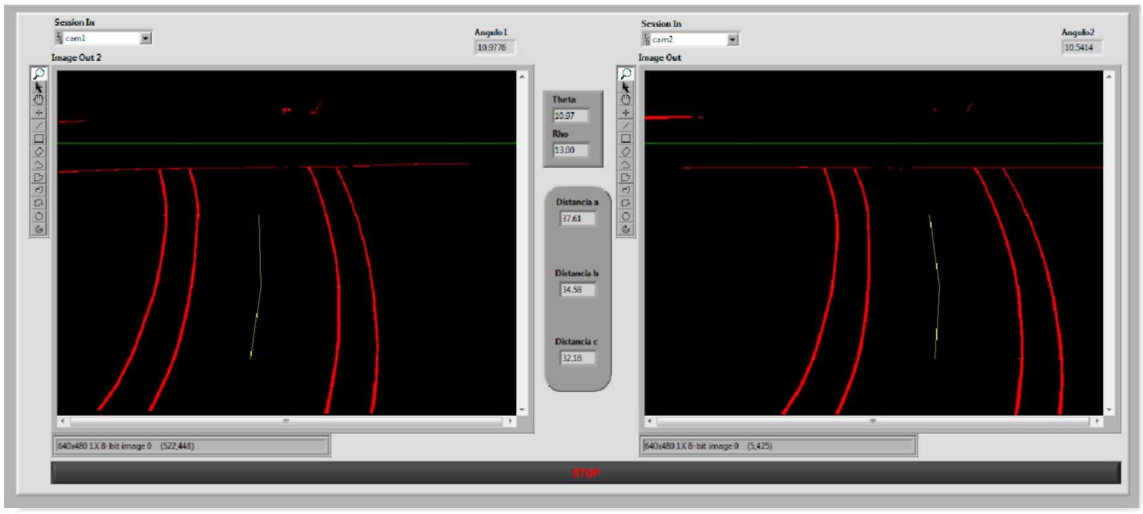


Figure 24: Front Panel of the vehicle on a curve with positive Theta.

According to the results in the screen, the road is detected exactly as it is. The angle was measured and corresponds to the angle Theta specified in the frame of each image.

In order to proove the equations of section 4.1 the tables 4 and 5 show the coordinates of detected edges in each of the two images analyzed. Each of the images detected 8 edges. Point $a_{1x}$, $b_{1x}$ and $c_{1x}$ are calculated as equation 19, 20 and 21 respectively, but with 8 elements.

$$a_{1x} = \left(\frac{289 + 289 + 322 + 326 + 492 + 496 + 531 + 535}{8}\right) = 410$$

$$b_{1x} = \left(\frac{275 + 275 + 309 + 313 + 497 + 501 + 536 + 539}{8}\right) = 405.625$$

$$c_{1x} = \left(\frac{261 + 264 + 297 + 300 + 503 + 506 + 540 + 540}{8}\right) = 401.375$$

Also points $a_{2x}$, $b_{2x}$ and $c_{2x}$ were calculated as follows,

$$a_{2x} = \left(\frac{169 + 172 + 202 + 205 + 365 + 368 + 402 + 405}{8}\right) = 286$$

$$b_{2x} = \left(\frac{156 + 159 + 189 + 193 + 370 + 373 + 407 + 410}{8}\right) = 281.75$$

$$c_{2x} = \left(\frac{144 + 147 + 177 + 180 + 375 + 378 + 412 + 415}{8}\right) = 278.5$$

And according to equation (1), the distances to the surface are calculated.

$$D_a = \frac{55 * -80}{286 - 410} = 35.48$$

$$D_b = \frac{55 * -80}{281.75 - 405} = 35.6$$

$$D_a = \frac{55 * -80}{278.5 - 401.375} = 35.8$$

Which correspond to figure 25.

Table 4: Edge detection data of the left image.

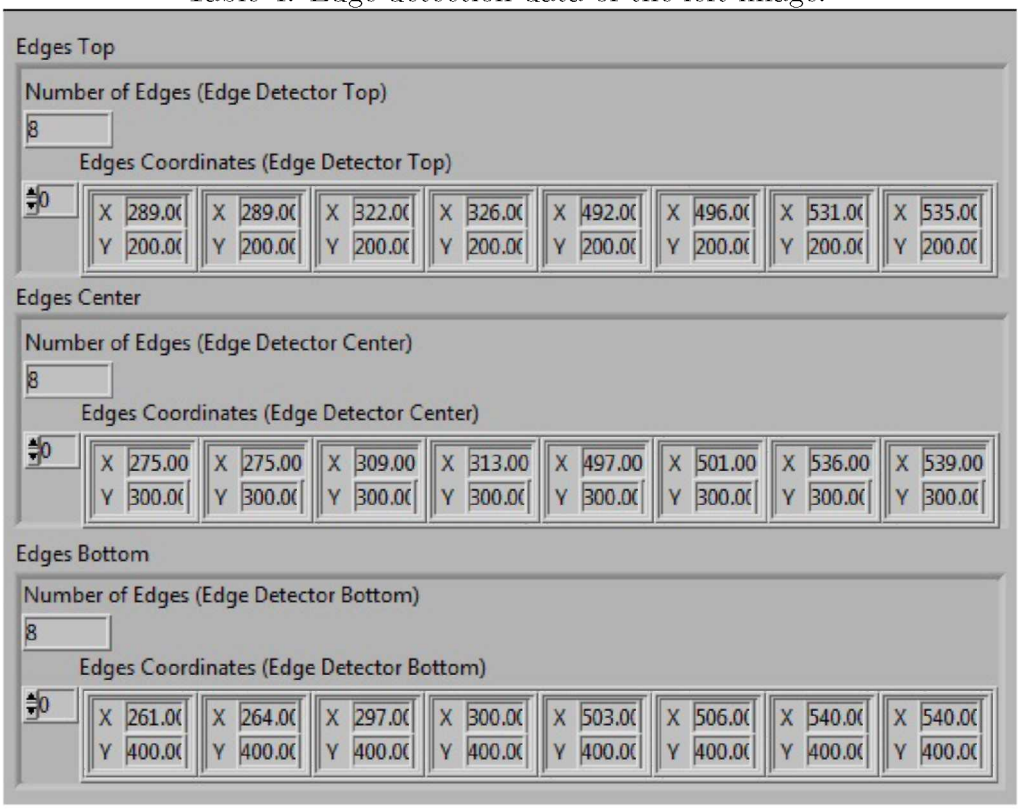

Table 5: Edge detection data of the right image.

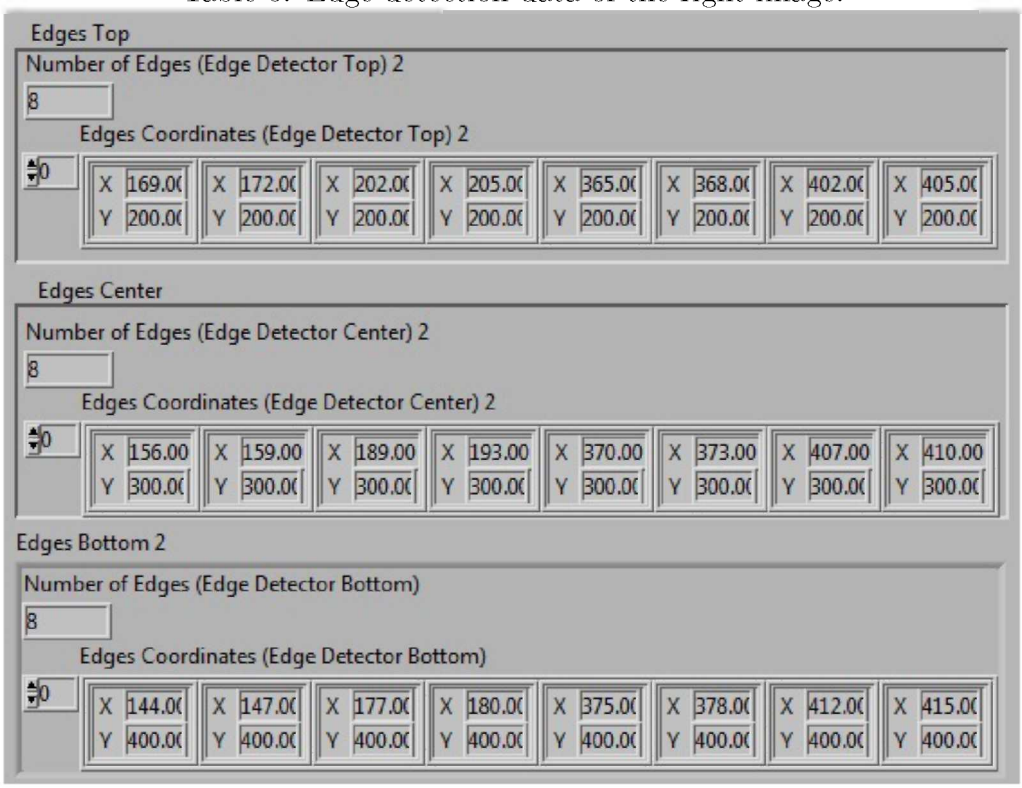



Figure 25: Results of the simulation.

To make sure that the distances calculated are the real distance from the camera to each point, the experiment explained in C.2 was carried out. The results are presented in table 6.

In order to use the stereo capabilities, the distance in the points a, b and c is analyzed continously. Since the road is always at the same distance from the cameras when no object is detected, when an object is placed in the road, the distance for that point changes significantly. Therefore, a change in the average distance detected in any of the points, will be considered as an obstacle, and the car will stop.

Table 6: Comparison of real and calculated distances

| Point | Real Distance | Calculated Distance |
|---|---|---|
| a | 108 cm. | 111 cm. |
| b | 73 cm. | 74.48 cm. |
| c | 55 cm. | 56.47 cm. |

Table 4 and 5 demonstrate that the edges are detected correctly. because the road is detected with 4 lines in that case, which correspond to 8 edges detected as illustrated in those tables.

According to table 6, it can be concluded that the maximum error in calculating the distance is at point a, with a 2.7% in error. The best match is at point b, with a 1.98% in error. The error at point c is 2.6%. Taking into account that the distances are compared with the same calculated distance in case of a detection of obstacle, the error in this distances doesn't influence the results, because they are relative. But it can be said that for this work, a 2.7% in error is good enough to complete the objective.

To illustrate the detection of obstacles, figure 26 shows the pair of images when no obstacle is detected.

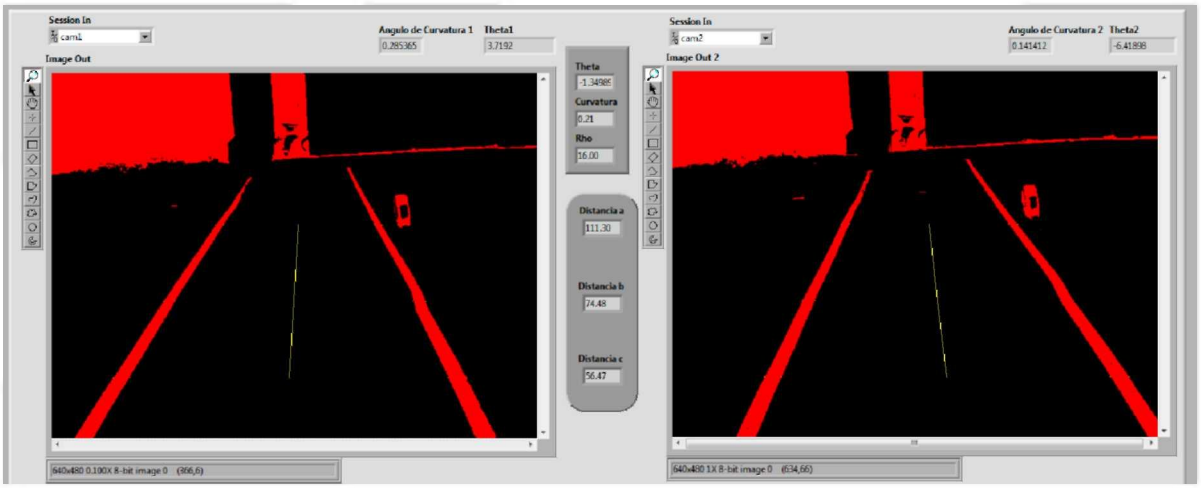


Figure 26: Image when no obstacle is detected

Figure 27 shows the case when an obstacle is present at point a. There is a change in the distance in point a and therefore the car stops.

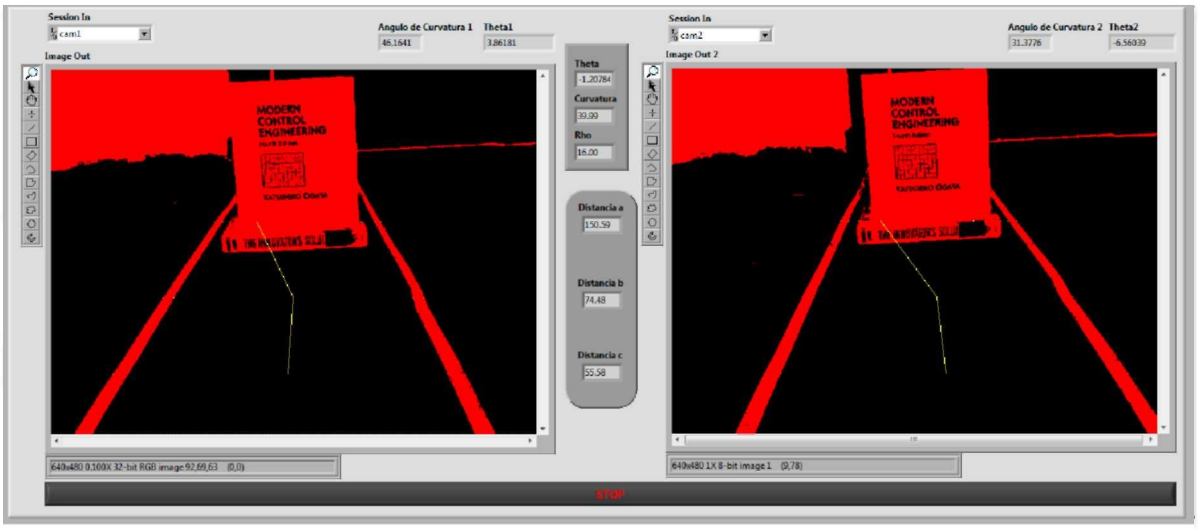


Figure 27: Detection of obstacle in point a

Figure 28 shows the case when an obstacle is present at point b. There is a change in the distance in point b and therefore the car stops.

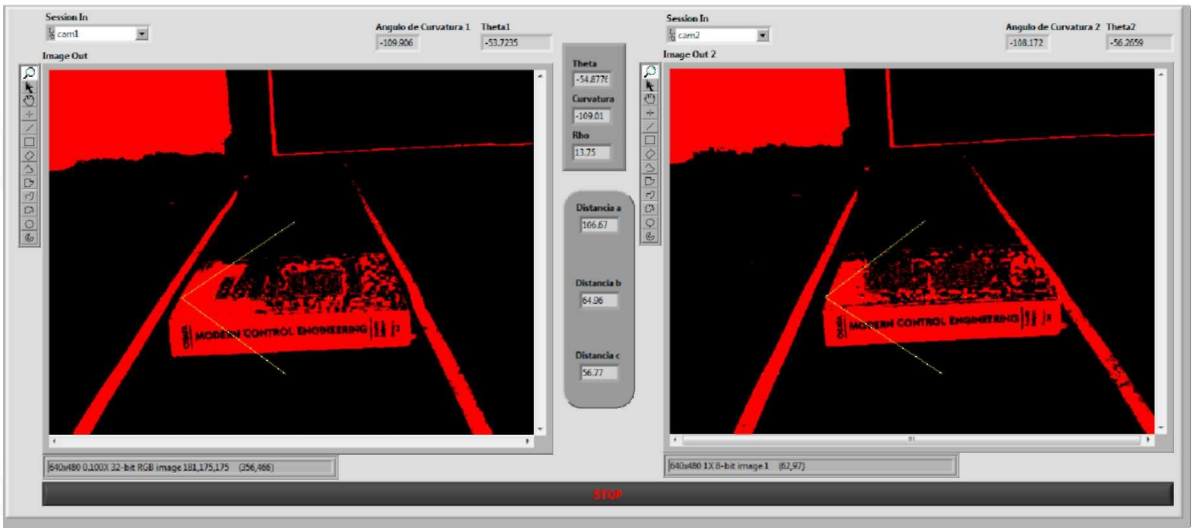


Figure 28: Detection of obstacle in point b

Figure 29 shows the case when an obstacle is present at point c. There is a change in the distance in point c and therefore the car stops.

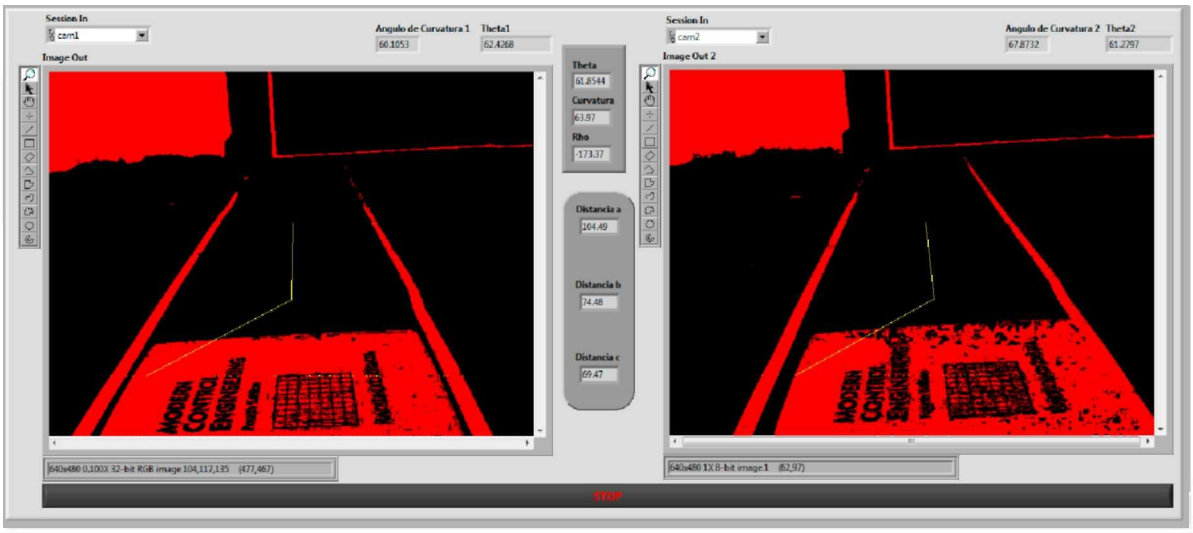


Figure 29: Detection of obstacle in point c

Figure 30 presents the case when an obstacle is located over the track at point b. There is a change in the distance in point b and therefore the car stops.

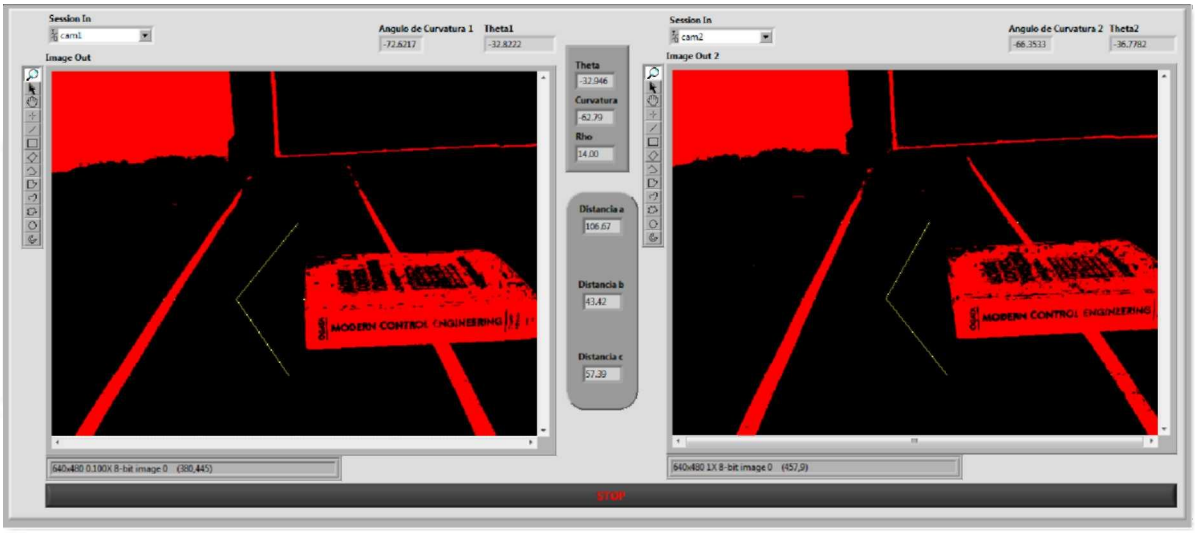


Figure 30: Detection of obstacle over the track in point b

The advantage of using stereovision is that the car does not need other sensors to obtain distances; in this case, stereovision allows the system to know the distance between the car and different points on the road, although it could be expanded in general to know the distances of the objects in the entire image. Therefore, stereovision is used to know the presence of obstacles on the road, without the use of any other type of sensor.

It can be concluded that the objects are detected in each of the frames and at each of the points a, b and c, no matter if the objects is between the road or above the track.

## 5.3 Fuzzy Controller

Using the table 7 as the rule base for the IF-THEN rules and the Mandani Implication, the fuzzy controller was implemented in the software.

Table 7: Rule Base.

| θ \ ρ | ++ | + | 0 | - | -- |
|---|---|---|---|---|---|
| ++ | -VB | -VB | -B | +S | +S |
| + | -VB | -B | -S | +S | +B |
| 0 | -B | -S | 0 | +S | +B |
| - | -B | -S | +S | +B | +VB |
| -- | -S | -S | +B | +VB | +VB |

Due to the predominant use of the triangular membership function in fuzzy control, the most economic functiom, the symmetric triangular shape was selected because it also consumes less computational processing. In figure 31-33 the membership function values for the input $\theta$, $\rho$, and the output, are presented respectively. This values were used for the fuzzy controller implementation. The values of the fuzzy sets where calculated according to the physical hardware specifications; while the angle cannot be greater than 50 degrees in a controllable situation, all the fuzzy sets where inside that extremes. The distance between the car's center and the center line of the road , in pixels, was seen to be as big as 200 pixels in each direction when the road was still visible for the two cameras, which is important to the road recognition. And for the output, the values for the singletons where based on the steering system, that has a range between -30 $^\circ$ and 30 $^\circ$.

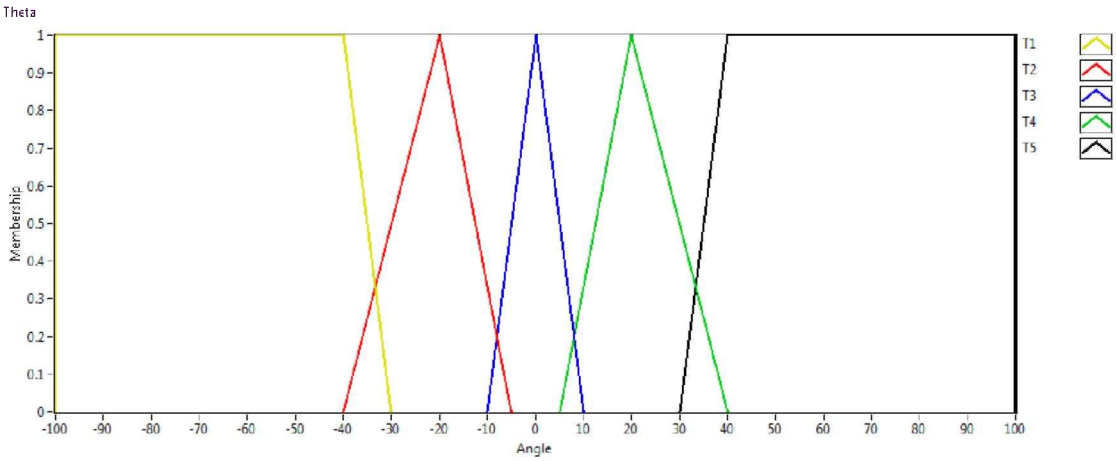


Figure 31: Membership functions of the input $\theta$.

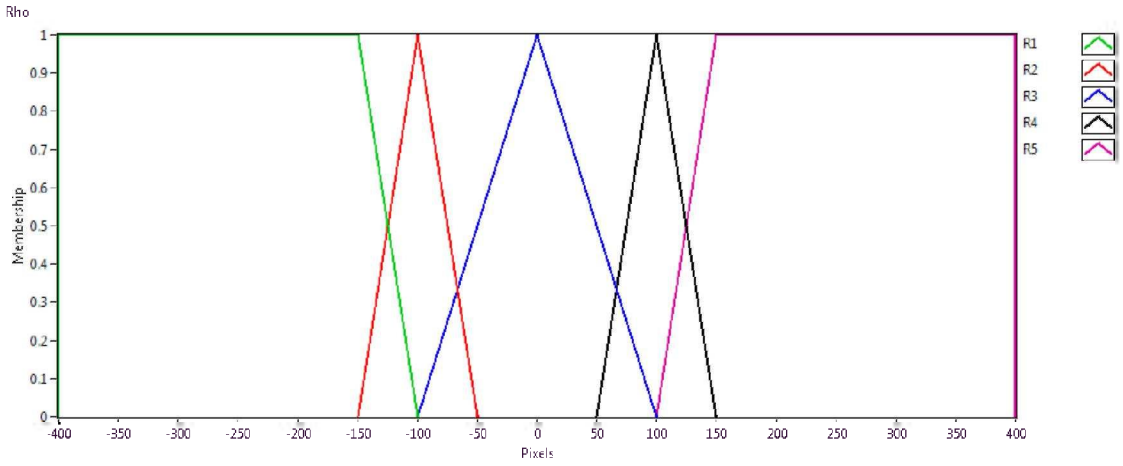


Figure 32: Membership functions of the input $\rho$.

Since a fuzzy singleton is a fuzzy set whose support is a single point in the space with a membership function of one, it is calculated much faster in comparisson with other methods of defuzzification, and that the reason it was used for this process.

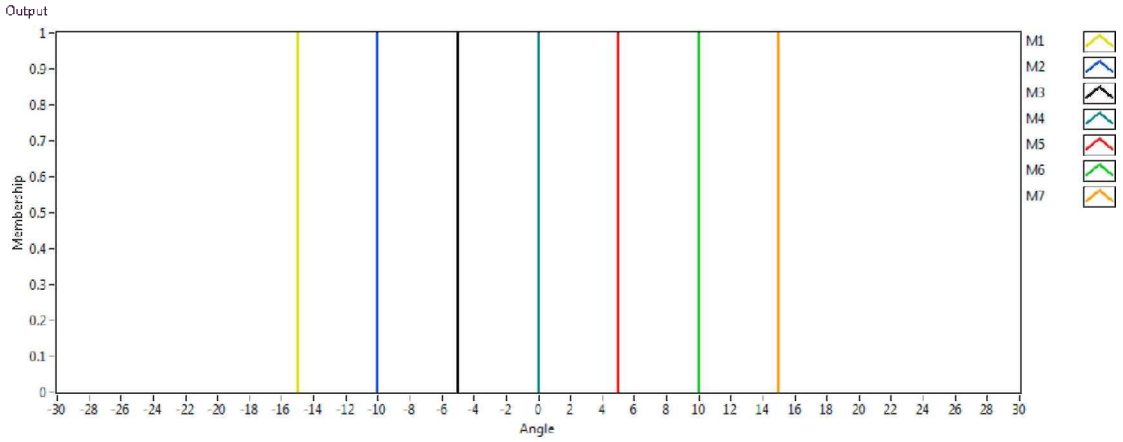


Figure 33: Membership functions of the output.

An experiment to verify the controllability of the car was carried out. The experiment is explained in appendix C.4. As a result, the SSE was calculated with the parameter $\rho$ to know the deviation from the central line of the track to the actual position of the car. SSE was used, because it penalizes big errors and doesn't take much into account the small errors. Table 8 shows the SSE for this trajectory.

Table 8: Sum of Squared Error

| | SSE |
|---|---|
| Autonomous robot | 3218 |

Figure 34 shows the graph of the manipulations done by the fuzzy controller, the error through the road and the parameter $\rho$. As seen in the figure, the car maintained its course controllable.

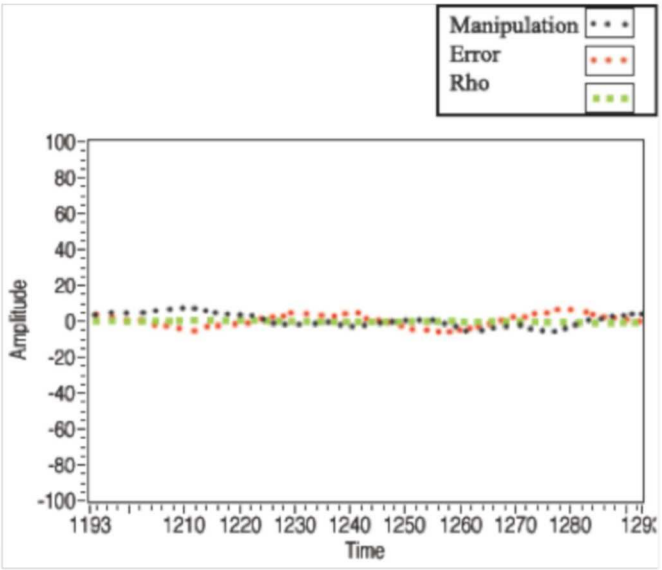


Figure 34: Manipulation, error and rho in car trajectory.

According to the SSE of 3218, it can be concluded that the average error in a time frame of 100 points is about 5.67 pixels per unit of time. The ideal SSE is 0, and considering that the road has a width of 500 pixels at point C, where rho is compared, the average error is about 1.134%. From figure 34, we can conclude that the maximum error is about 10 pixels; also taking a width of 500 pixels, the maximum error is about 2%. With this results, it can be concluded that the car maintains its position through the road with a good performance; an average error of 1.13% and a maximum error of 2%.

## 5.4 Software Implementation

For the software integration, the program LabVIEW was used to develop the code needed to do the necessary tasks: grab image from the cameras, make an image processing to detect the road, the fuzzy control calculations and finally, the comunication with the actuators. The procedure that is done continously is presented in figure 35, and the explanation of the entire code is shown in appendix B.

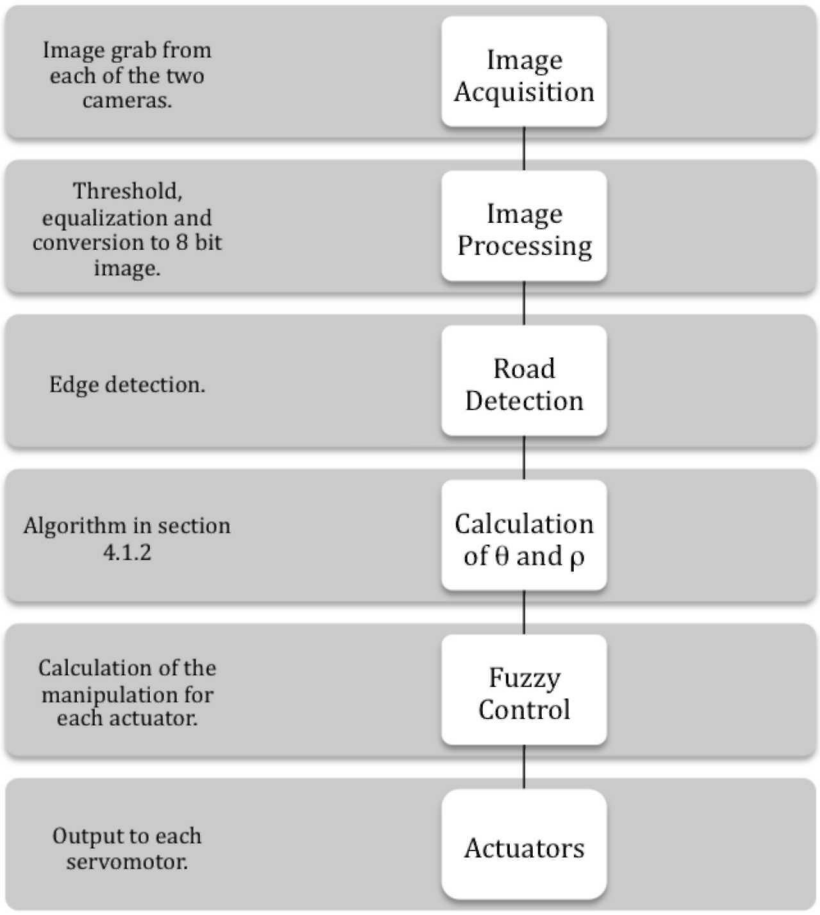


Figure 35: Flowchart of the software solution.

It can be concluded that LabVIEW is a good software for this kind of implementations because it is able to integrate the software and the hardware easily. In this case the communication with the serial data was possible, and also the image acquisition for the stereocamera. Also the graphical programming used was fundamental to make a very understandable and configurable code for this objective.

# 6 Conclusions

The work done in this thesis was to develop an autonomous vehicle capable of following a road with the method proposed. In order to do this, a fuzzy controller was designed, stereovision capabilities were implemented and also the image processing to detect the road.

An intelligent robotic vehicle is successfully developed. The stereocamera installed on the vehicle is able to detect the road, no matter if it is curved or straight. A fuzzy controller is developed for the autonomous vehicle. An analysis and design of fuzzy control laws for steering control of the nonholomic robotic vehicle are presented. The image processing algorithm designed is presented. The desired steering angle is given by the stereocamera and the steering angle of the front wheels of the robotic vehicle is controlled by the proposed fuzzy controller. Experiments demonstrate that the vehicle with the proposed fuzzy controller and image processing algorithm can automatically follow the road.

This study comprehends the design of a simple but practial image processing algorithm, the design of a fuzzy controller with the characteristics needed for this autonomous robot. We observed that the vehicle can be driven autonomously in the road proposed. This is the base for further improvements.

As thought before, the most important part of the system is the stereovision camera and its image processing. The speed of image grabbing is necessary for this kind of applications where the vehicle can travel fast, also the image processing is fundamental to detect the road on time. It can be said, that this solution for the objective proposed is cheap, and would be feasible to implement on other applications.

# 7 Further Work

Implement the obstacle avoidance in order to not only stop the car if an obstacle is detected, but also to evade the obstacle and keep the car inside the road. In this thesis, the obstacles are only detected but the autonomous vehicle doesn't respond to them. To do this, the controller should be enhanced.

Use more sensors for the robot like: GPS or Lasers. GPS could help the car to know the road in advance and apply a different control method. Lasers could be useful for detecting obstacles at a farther distance than the stereocamera. In general, the use of more sensors could help to better drivability of the vehicle. In this thesis, the stereovision is the only sensor used for the vehicle. It detects the road and deepness, but more sensors would be necessary for more capabilities for the robot.

Implement this algorithm to a real car in order to expand its applications. Although this robot could be compared to a real car, the real applications are in the car industry and would challenge the algorithm proposed. Also, improvements in the fuzzy controller should be analyzed to make it more stable and robust to different situations. The adjustments of the image processing could be automatized in order to adapt to different situations automatically.

# References


[1] *The Concise Handbook of Real-Time Systems.* TimeSys Corporation, 2002.

[2] Robotics fundamentals series: Stereovision., December 2008.

[3] Theo; Lopez Antonio M. Alvarez, Jose M.; Gevers. 3d scene priors for road detection. *Computer Vision and Pattern Recognition*, pages 57–64, 2010.

[4] A.OlivaandA.Torralba. Modelingtheshapeofthescene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[5] S. Baluja. Evolution of an artificial neural network based autonomous land vehicle controller. *Transaction on Systems, Man, and Cybernetics*, pages 450–463, 1996.

[6] L. Beji and Y. Bestaoui. An adaptive control method of automated vehicles with integrated longitudinal and lateral dynamics in road following. *Proceeding of The Second International Workshop on Robot Motion and Control*, pages 201–206, October 2001.

[7] R.; Aubert D.; Glaser S. Benmansour, N.; Labayrade. Stereovision-based 3d lane detection system: a model driven approach. *Intelligent Transportation Systems*, pages 182–188, 2008.

[8] R.; Aubert D.; Glaser S. Benmansour, N.; Labayrade and D. Gruyer. A model driven 3d lane detection system using stereovision. *Control, Automation, Robotics and Vision*, pages 1277–1282, 2008.

[9] C.; Fedriga R.I.; Grisleri P. Broggi, A.; Caraffi. Obstacle detection with stereo vision for off-road vehicle navigation. *Computer Vision and Pattern Recognition*, 2005.

[10] M. Zhang C. Ding, G. Cui and P. Duan. Mobile robot's road following based on color vision and fpga control strategy. *Proceeding of The Sixth World Congress on Intelligent Control and Automation*, pages 3124–3128, 2006.

[11] G.; Bensrhair A. Cabani, I.; Toulminet. A fast and self-adaptive color stereo vision matching; a first step for roa ostacle detection. *Intelligent Vehicles Symposium*, pages 58–63, 2006.

[12] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI- 8, No. 6, November 1986.

[13] Zezhong Xu ; Yanbin Zhuang ; Huahua Chen. Obstacle detection and road following using laser scanner. *Intelligent Control and Automation*, pages 8630–8634, October 2006.

[14] G.E.; Koayashi K.; Overholt J.L.; Lescos P. Choek, K.C.; Smid. A fuzzy logic intelligent control system architecture for an autonomous leader-following vehicle. *American Control Conference*, 1:522–526, 1997.

[15] D Chunzhao Guo; Mita, S.; McAllester. Stereovision-based road boundary detection for intelligent vehicles in challenging scenarios. *Intelligent Robots and systems*, 2009.

[16] A. A. Efros D. Hoiem and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[17] S. H. Han M. H. Lee D. Y. Jeong, S. J. Park and T. Shibata. A study on neural networks for vision-based road following of autonomous vehicles. *International Symposium on Industrial Electronics*, pages 1609–1614, 2001.

[18] S. Danescu, R.; Nedevschi. Adaptive and robust road tracking system based on stereovision and particle filtering. *Intelligent Computer Communication and Processing*, pages 67–73, 2008.

[19] O. ; Zhiwei Zhu ; Samarasekera S. ; Kumar-R. Das, A. ; Naroditsky. Robust visual path following for heterogeneous mobile platforms. *Robotics and Automation*, pages 2431–2437, 2010.

[20] Jose E. Naranjo; Miguel A. Sotelo; Carlos Gonzalez; Ricardo García; Teresa de Pedro. Using fuzzy logic in automated vehicle control. *Intelligent Sytems*, pages 36–45, 2007.

[21] Alessandro Saffiotti Dimiter Driankov. Fuzzy logic techniques for autonomus vehicle navigation. *Physica-Verlag*, 2001.

[22] Dimiter Driankov. *An introduction to fuzzy control.* Springer-Verlag, 1996.

[23] Hellendoorn H. Driankov D. and Reinfrank M. *An Introduction to Fuzzy Control.* Springer-Verlag, 1996.

[24] W.; van den Heuvel J.C.; Groen F.C.A. Dubbelman, G.; van der Mark. Obstacle detection during day and night conditions using stereo vision. *Intelligent Robots and Systems*, pages 109–116, 2007.

[25] Pedro Espinosa Perez. *Desarrollo de una Metodología para la Integración de Sistemas Robot-Visión.* Instituto Tecnológico y de Estudios Superiores de Monterrey, 2006.

[26] A. Ewald and V. Willhoeft. Laser scanners for obstacle detection in automotive applications intelligent vehicles symposium. *Proceedings of the IEEE on Intelligent Vehicle*, pages 682–687, 2000.

[27] R.I.; Ghidoni S. Fascioli, A.; Fedriga. Vision-based monitoring of pedestrian crossings. *Image Analysis and Processing*, pages 566–574, 2007.

[28] D. Hoiem. Seeing the world behind the image: Spatial lay- out for 3d scene understanding. Master's thesis, Carnegie Mellon, August 2007.

[29] S. ; Boehling R. ; Camoriano N. ; Cardwell-W. ; Jannaman G. ; Purcell A. ; Ross D. ; Russel E. Hong, D. ; Kimmel. Development of a semi-autonomous vehicle operable by the visually-impaired. *Multisensor Fusion and Integration for Intelligent Systems*, pages 539–544, August 2008.

[30] T. Gevers J. Alvarez and A. Lopez. Vision based road detection using road models. *ICIP*, pages 2073–2076, 2009.

[31] L. A. L Nozal J. L Arroyabe, G. Aranguren and J. L. Martin. Autonomous vehicle guidance with fuzzy algorithm. *Annual Conference of the IEEE Industrial Electronics Society*, pages 1503–1508, 2000.

[32] A. Torralba S. Avidan J. Sivic, B. Kaneva and W. T. Free-man. Creating and exploring a large photorealistic virtual space. *Workshop on Internet Vision*, June 2008.

[33] F. ; Wollherr D. ; Buss M. Lidoris, G. ; Rohrmuller. The autonomous city explorer (ace) project — mobile robot navigation in highly populated urban environments. *Robotics and Automation*, pages 1416–1422, May 2009.

[34] A. Broggi M. Bertozzi and A. Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robot Automat System*, 32:1–16, 2000.

[35] M. Cellario A. Fascioli P. Lombardi M. Bertozzi, A. Broggi and M. Porta. Artificial vision in road vehicles. *Proceedings of the IEEE Special Issue on Visual Perception*, pages 1258–1271, 2002.

[36] E. H. Mandani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. Journal of Man-machine Studies*, 1(1), 1975.

[37] T. Marita. Barriers detection method for stereovision-based acc systems. *Intelligent Computer Communication and Processing*, pages 95–101, 2009.

[38] L. Montano and JR Asensio. Real-time robot navigation in unstructured environments using a 3d laser rangefinder. *Int. Conf. on Intelligent Robots and Systems*, pages 526–532, 1997.

[39] C. Neagoe, V. ; Tudoran. Road following for autonomous vehicle navigation using a concurrent neural classifier. *Automation Congress*, pages 1–6, September 2008.

[40] Hung T Nguyen. *A first course in fuzzy logic.* CRC Press, 1997.

[41] Seung-Hun Kim ; Chi-Won Roh ; Sung-Chul Kang ; Min-Yong Park. A hybrid autonomous / teleoperated strategy for reliable mobile robot outdoor navigation. *SICE-ICASE*, pages 3120–3125, 2006.

[42] R. ; Royere C. ; Hautiere N. ; Aubert-D. Perrollaz, M. ; Labayrade. Long range obstacle detection using laser scanner and stereovision. *Intelligent Vehicles Symposium*, pages 182–187, September 2006.

[43] C. Mertz C. Thorpe C. Wang R. Aufrere, J. Gowdy and T. Yata. Perception for collision avoidance and autonomous driving. *Mechatronics*, 13(10):1149–1161, December 2003.

[44] E. Messina R. Madhavan and J. Albus. *Intelligent Vehicle Systems A 4D/RCS Approach.* New York: Nova Science Publishers Inc., 2006.

[45] Richard E. Woods Rafael C. Gonzalez. *Digital Image Processing.* Addison-Wesley, 1996.

[46] P.; Lopez Larios F.A Ramirez, J.M.; Gomez-Gil. Robot-visionsystem for autonomous vehicle navigation with fuzzy-logic control using labview. *Electronics, Robotics and Automotive Mechanics Conference*, pages 295–302, 2007.

[47] Peter E. Hart Richard O. Duda. Use of the hough transform to detect lines and curves in pictures.

[48] J. H. Lee S. Y. Oh and D. H. Choi. A new reinforcement learning vehicle control architecture for vision-based road following. *Transactions on Vehicular Technology*, pages 997–1000, May 2000.

[49] Linda Shapiro and George Stockman. *Computer Vision.* Prentice-Hall, 2001.

[50] Yaling Du; Xiaoying Gao; Zhun Liu; Mei Sun. The new navigation system for automatic guided vehicle. *Control and Decision Conference*, pages 4653–4658, 2008.

[51] S. J. Chang T. H. S. Li and Y. X. Chen. Implementation o f human-like driving skills by autonomous fuzzy behavior control on an fpga-based car-like mobile robot. *Transactions on Industrial Electronics*, pages 867–880, October 2003.

[52] A. Torralba and P. Sinha. Statistical context priming for object detection. *ICCV*, page 763, 2001.

[53] J.C.; Groen F.C.A. van der Mark, W.; van den Heuvel. Stereo based obstacle detection with uncertainty in rough terrain. *Intelligent Vehicles Symposium*, pages 1005–1012, 2007.

[54] M.E. Yi Fu ; Li, H. ; Kaye. Hardware/software codesign for a fuzzy autonomous road-following system. *Systems, Man, and Cybernetics*, pages 690–696, 2010.

[55] M Yi Fu; Li, H.; Kaye. Design and stability analysis of a fuzzy controller for autonomous road following. *Intelligent Vehicle Symposium*, 66-71, 2009.

[56] A. M. Zhang and R. A. Russell. Dominant orientation tracking for path following. *IRSJ*, pages 3885–3889, 2005.

[57] Liu Jin-dong Zhang Hai-bo, Yuan Kui. A fast and robust vision system for autonomous mobile robots. *Robotics, Intelligent Systems and Signal Processing*, 1:60–65, October 2003.

[58] Zezhong Xu Zhiyu Xiang and Jilin Liu. Small obstacle detection for autonomous land vehicle under semi-structural environment. *Proceedings of IEEE Intelligent Transportation Systems*, 1:293–298, 2003.

# Part II

# Appendixes

# A Hardware Specifications

## A.1 StereoCamera

The StereoCamera was built using two Microsoft LifeCam VX-2000. They were attached so that the cameras are parallel to each other and with a distance of 55 millimeters in order to see clearly the road.


Figure 36: Webcam.

The main features are:

- Captures up to 30 frames per second, which means the video images you see are smooth and seamless.
- Built-In microphone.
- 3X Digital Zoom.
- VGA Video Sensor: Clear VGA video and sharp 1.3 megapixel (interpolated) still photos. Camera auto-adjusts for low-light conditions for the best video possible.
- USB 2.0 Connection.

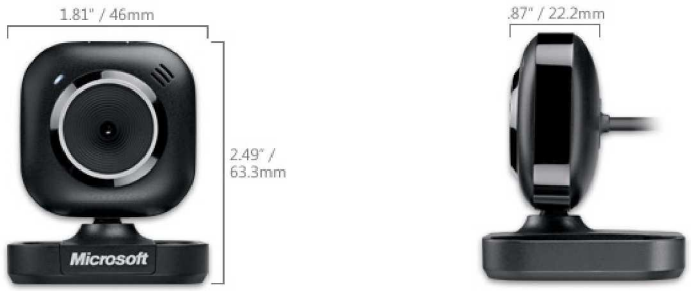


Figure 37: Dimensions of the webcam.

This camera was selected because of its small size, quality of image and mainly, because the 30 fps it can grab; this amount of fps is needed for the controllability of the car. The USB 2.0 connection is important to grab the images faster from the camera to the computer. An important consideration, is that the drivers of this webcam allow Windows to recognize two different cameras when two equal cammeras are connected to the computer. It is very important, if it is planned to use both cameras to used them as a stereocamera.

## A.2 Servomotors

Each of the two servomotors used is an Standard Servo HS-422 sold by Lynxmotion.

Figure 38: Servomotor HS-422.

The main specifications are:

- Control System: +Pulse Width Control 1500uSec Neutral.
- Operating Voltage Range: 4.8V to 6.0V
- Operating Speed: 0.21 sec/60° at no load at 4.8V and 0.16 sec/60° at no load at 4.8V.
- Stall Torque: 3.3 kg.cm at 4.8V and 4.1 kg.cm at 6.0V
- Operating Angle: 45° /one side pulse traveling 400usec.
- Direction: Clockwise/pulse traveling 1500 to 1900 usec.
- Dead band width: 8usec.
- Dimensions: 40.6x19.8x36.6 mm.
- Weight: 45.5 g

## A.3 Servo controller card

In order to controll the servomotors, a servo controller SSC-32 from Lynxmotion was used.

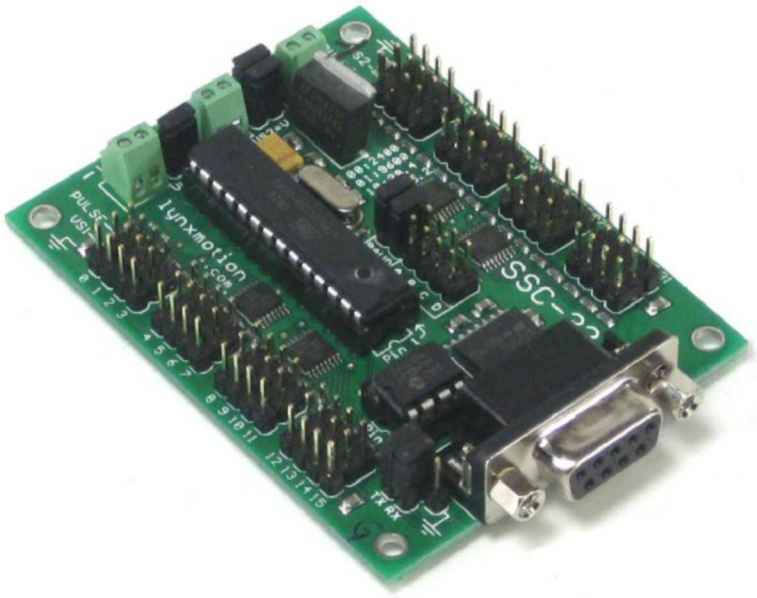


Figure 39: Servo Controller SSC-32.

The SSC-32 (serial servo controller) is a small preassembled servo controller with some big features. It has high resolution (1uS) for accurate positioning, and extremely smooth moves. The range is 0.50mS to 2.50mS for a range of about 180°. The motion control can be immediate response, speed controlled, timed motion, or a combination. A unique "Group Move" allows any combination of servos to begin and end motion at the same time, even if the servos have to move different distances. This is a very powerful feature for creating complex walking gaits for multi servo walking robots. The servo's position or movement can be Queried to provide feedback to the host computer. There is even a 12 servo Hexapod sequencer built in. This allows complete control of all aspects of the alternating tripod gait simply by transferring a few values from the host controller. Any output can be used as a TTL level output. There are 4 digital inputs that are static or latched, so you don't have to worry about missing a short event. They can also be used as analog inputs. There are three terminal blocks for powering options. The DB9 input has true RS-232 levels for use with a PC.

The specifications of this card are:

- Microcontroller = Atmel ATMEGA168-20PU
- EEPROM = 24LC32P (Required for 2.01GP)
- Speed = 14.75 MHz
- Internal Sequencer = 12 Servo Hexapod (Alternating Tripod)
- Serial input = True RS-232 or TTL, 2400, 9600, 38.4k, 115.2k, N81
- Outputs = 32 (Servo or TTL)
- Inputs = 4 (Static or Latching, Analog or Digital)
- Current requirements = 31mA
- PC interface = DB9F
- Microcontroller interface = Header posts
- Servo control = Up to 32 servos plug in directly
- Servo type supported = Futaba or Hitec
- Servo travel range = 180° Servo resolution = 1uS, .09°
- Servo speed resolution = 1uS / Second
- Servo motion control = Immediate, Timed, Speed or Synchronized.
- PC board size = 3.0" x 2.3"
- VS current capacity = 15 amps per side, 30 amps max

## A.4 Power Module of the Speed Control

The speed of the car is controlled as shown in section 5.1.1. The power module was implemented with a PWM signal from the microcontroller; the PWM switches on and off a transistor in order to regulate the voltage in the DC motor. Pulse-width modulation is an effective method for adjusting the amount of power delivered to an electrical load. Figure 40 shows the schematic diagram of this speed controller.

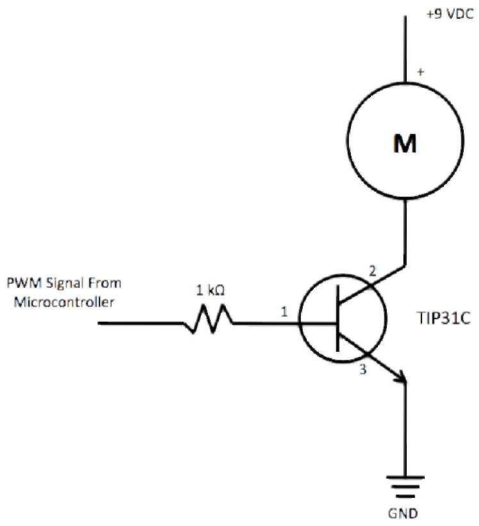


Figure 40: Power module for the speed control.

## A.5 Arduino Microcontroller

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller.

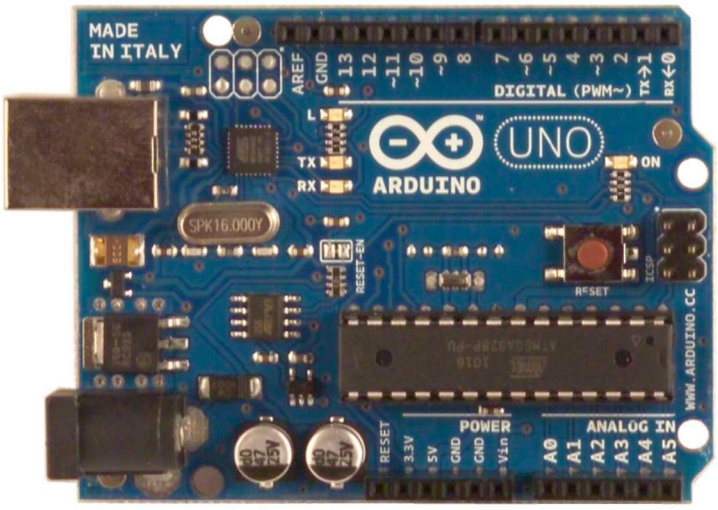


Figure 41: Front view of the Arduino Uno.

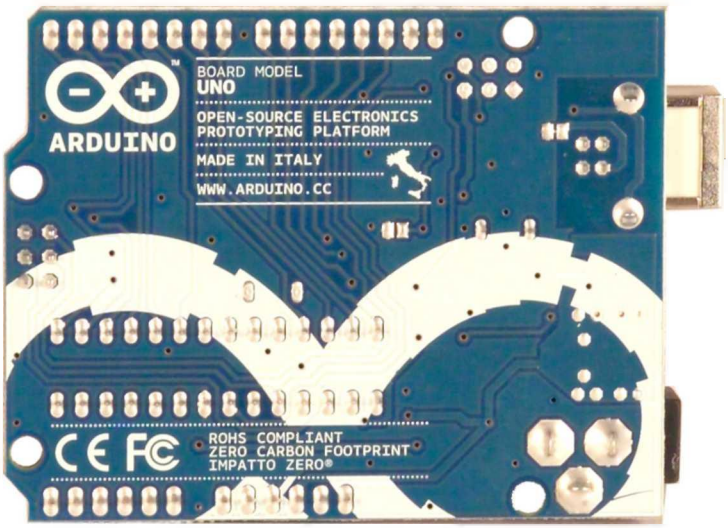


Figure 42: Back view from the Arduino Uno.

The specifications of the board are:

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MHz

## A.6 Laptop

A laptop was used in order to do all the image processing, the fuzzy control, and to send the manipulations to the actuators. A big consideration for this task was that it should be small and have at least two USB connections. The laptop is a Dell Inspiron 300M.


Figure 43: Dell Inspiron 300m.

Main specifications:

- Processor: Intel Pentium M 1.2 GHz
- RAM: 1 Gb
- Dimensions: 10.8x9.2x0.1 in
- Weight: 3.0 lbs
- Graphics: Intel Extreme Graphics

# B Program Description

The program used to implement this thesis is LabVIEW 2009. It is a platform and development environment for a visual programming language from National Instruments. The graphical language is named "G". LabVIEW is commonly used for data acquisition, instrument control, and industrial automation. One benefit of LabVIEW over other development environments is the extensive support for accessing instrumentation hardware. Drivers and abstraction layers for many different types of instruments and buses are included or are available for inclusion. These present themselves as graphical nodes. The abstraction layers offer standard software interfaces to communicate with hardware devices. The provided driver interfaces save program development time. The sales pitch of National Instruments is, therefore, that even people with limited coding experience can write programs and deploy test solutions in a reduced time frame when compared to more conventional or competing systems. A new hardware driver topology (DAQmxBase), which consists mainly of G-coded components with only a few register calls through NI Measurement Hardware DDK (Driver Development Kit) functions, provides platform independent hardware access to numerous data acquisition and instrumentation devices.

There are two types of windows when programming in LabVIEW. The first is the front panel, where the HMI presents the results, and graphs that the user wants to show. The other is the block diagram, where the graphical programming is done.

## B.1 Image Processing

In the front panel developed, there are mainly 5 sections: edge detection, threshold, equalization, stereovision and configuration of stereocamera. This sections can be modified in order to let the user modify the parameters for special enviroments.

In edge detection section shown in Figure 44, the three lines that are constantly detecting edges are presented. In the top, center and bottom box, the edges detected by the top, center and bottom edges are presented. The values are shown in pixels, with coordinates in the plane $(X, Y)$. The number of edges detected are also presented.

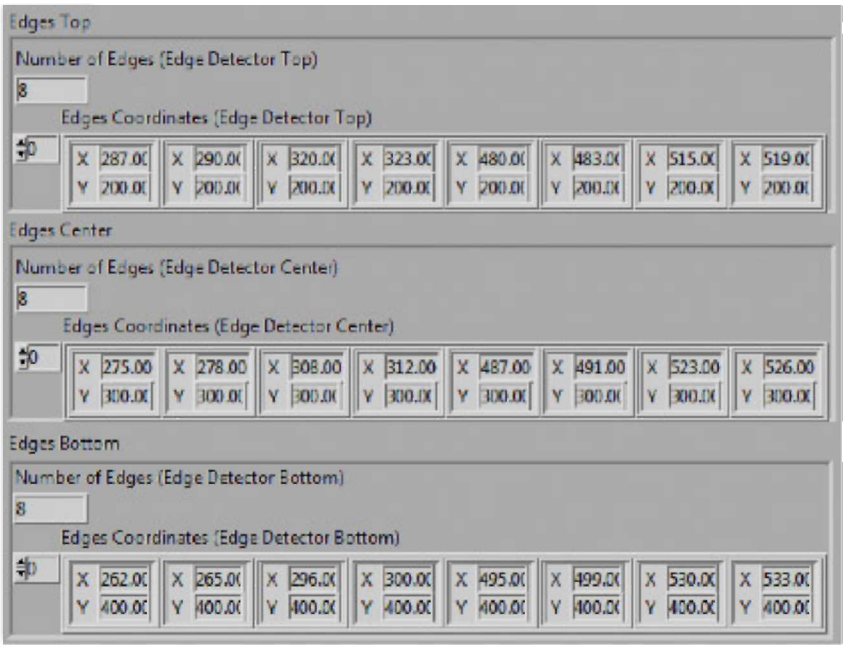


Figure 44: Edge detection section in front panel.

In the threshold section, the configuration of the theshold in the three planes of the RGB plane is presented. Because it has an 8-bit resolution, 256 values can be introduced. Modifying this values will determine which colors can be seen in the image, and these will be introduced to the image processing part of the program. In Figure 45, the threshold section is shown.

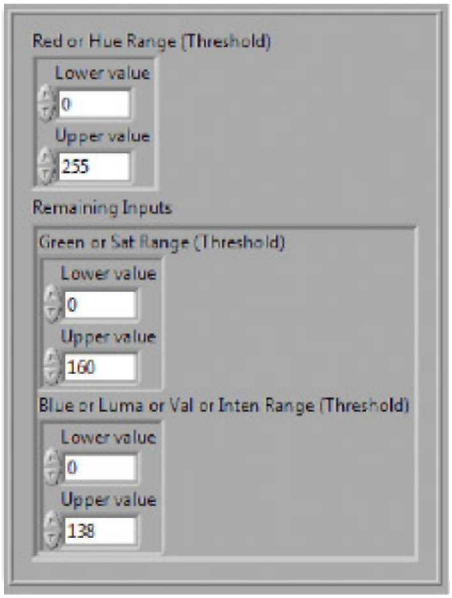


Figure 45: Threshold section in front panel.

The equalization section is important. In this part, the values for the brightness,

contrast and gamma are modified for each of the three planes in the RGB color scheme. Modifying this values will determine the brightness of the scene, the contrast and the gamma in order to identify the values according to the enviroment. Figure 46 shows this section in the front panel.

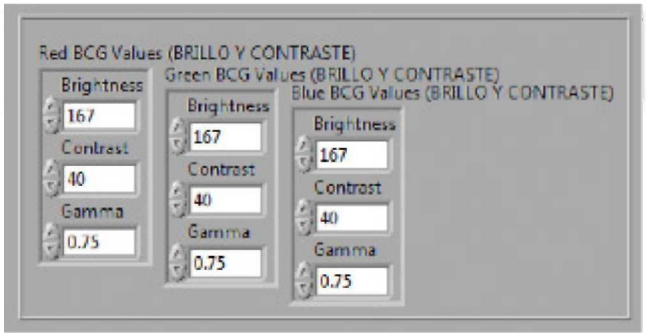


Figure 46: Equalization section in front panel.

The images that are grabbed by the stereocamera are shown in the stereovision section. The left image is the image grabbed by the right camera, and the right image the one grabbed by the left camera. This images are already processed by the image processing part of the program. Basically, the red lines represent the lines that are important in the program. The yellow lines are the detection of the central line according to procedure presented in section 4.1.2. Between these two images, the distance between the camera and the points a, b and c is shown in centimeters. $\theta$ and $\rho$ are shown above these distances, in degrees and pixels respectively. Above each of the images, the camera wanted to grab the images can be selected. Below the images, the stop botton for the program is located. Figure 47 shows the stereovision section.

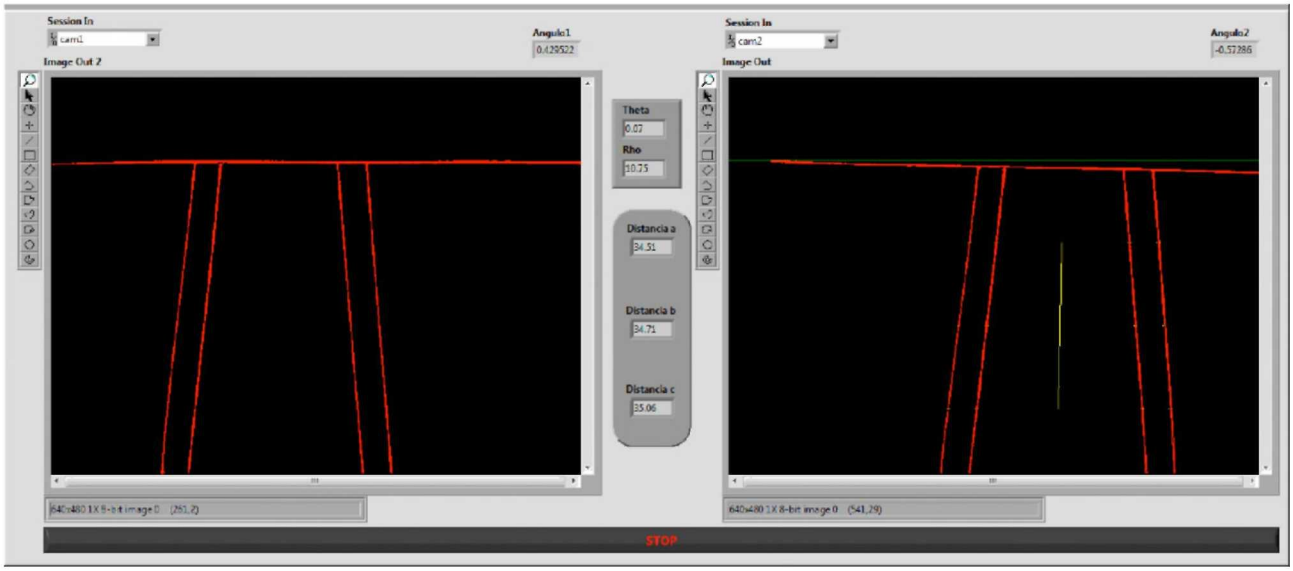


Figure 47: Stereovision in front panel.

In order to detect distances, there are two parameters that are needed. The first is the distance between the cameras, and the other, the focal length. This parameters are shown in the front panel in order to adjust them. Figure 48 shows this parameters.

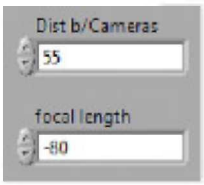


Figure 48: Configuration of stereocamera in front panel.

All this sections are shown in Figure 49. This is the entire view of the front panel.

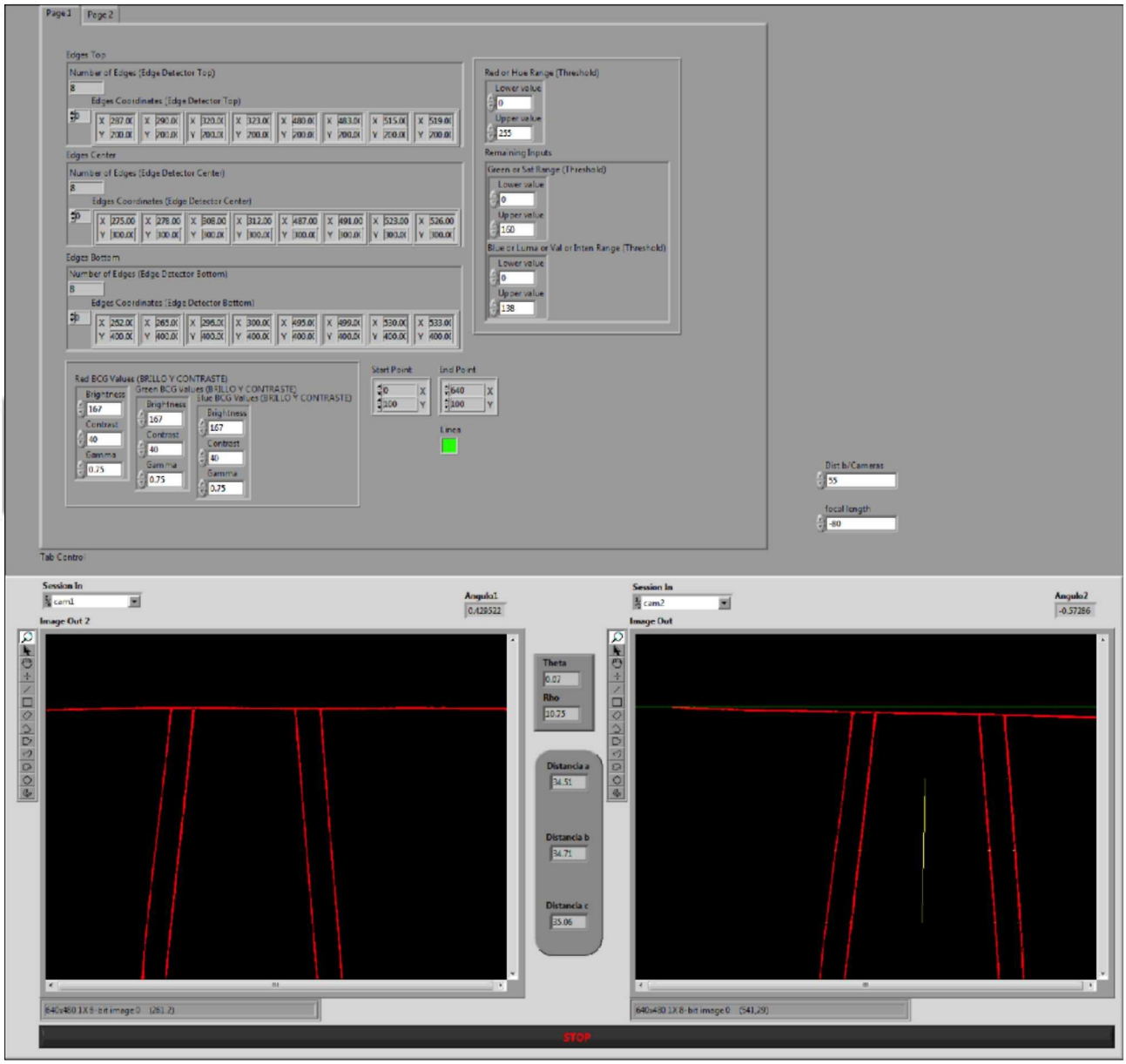


Figure 49: Front panel of the program.

The code is structured according to the process in section 4.1. First a session is opened and configured in each webcam. Then the process of equalization is done, then the thresholding, the edge detection in cascade for each of the lines (top, center and bottom). After that, an overlay of the images is done. The procedure in section 4.1.2 is programmed inside a mathscript structure. Another overlay of the lines a-b, and b-c is done. Finally, the calculation of the distances in points a, b and c is done. And also rho and theta are calculated and sent to the fuzzy controller.

The image acquisition is started with the name of the camera. It opens the usb camera, and the configuration of some parameters can be done in this section. Also a blank image is created in memory to store the grabbed images. Figure 50 shows this

part of the block diagram.

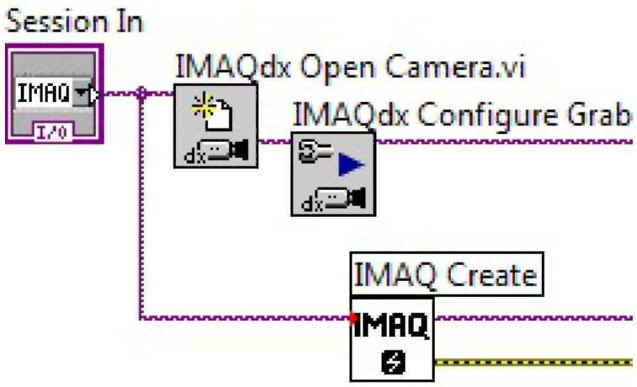


Figure 50: Configuring image acquisition

After acquiring the image, two processes are done; first, the equalization of the image with the values of BCG(Brightness, Contrast and Gamma) that que user enters in each RGB(Red, Green, Blue) plane. Then, a threshold is applied to the equalized image. The threshold is applied in each of the three RGB planes also. The values for the threshold are also entered by the used in the front panel. Figure 51 shows the equalization and thresholding parts of the block diagram.

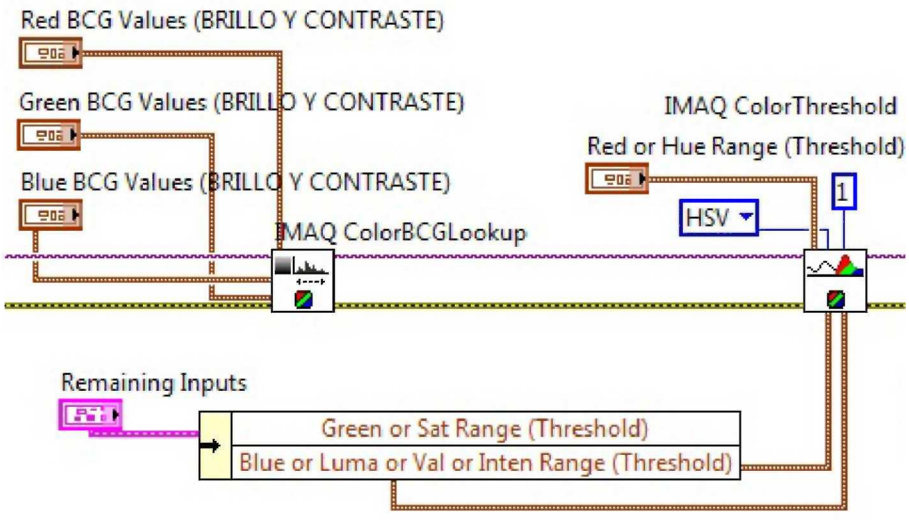


Figure 51: Equalization and threshold

Following the threshold and equalization of the image, the first edge detection is done. This is done by reading only a single line of pixels. Each line (Top, Center and

Bottom) have the same process. First, the ROI (Region of Interest) is configured, it contains the coordinates of the line to be analyzed. The edge detection is done by two subVIs. The outputs are the coordinates of the edges detected and the number of edges detected. Figure 52 shows the edge detection section.

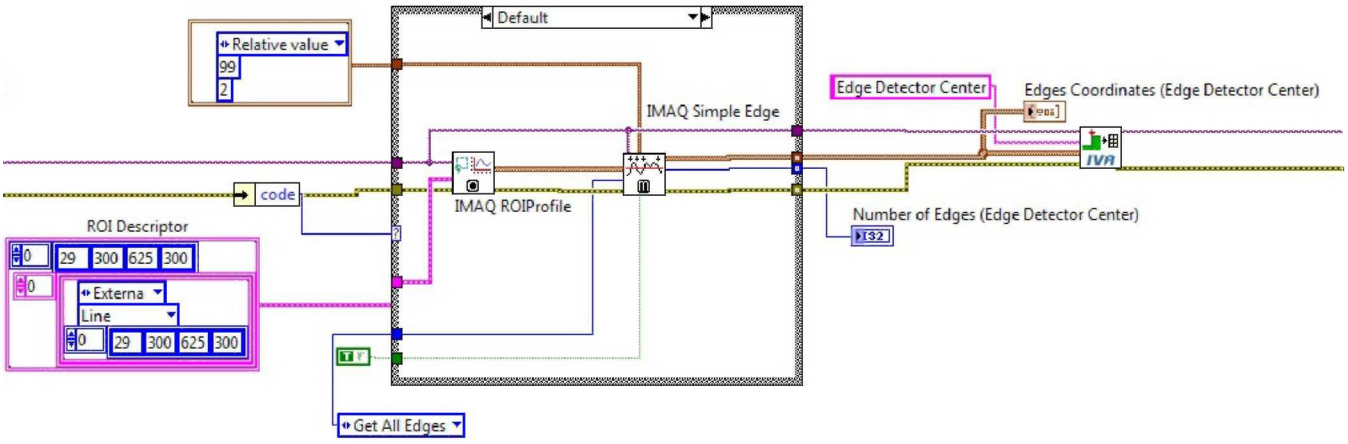


Figure 52: Edge detection

After the three lines are analyzed for edge detection, an overlay of those lines is done in order to show the edges detected in the front panel. The coordinates of the edges detected is read by a Mathscript structure that calculates the points a, b and c. Also the angle of curvature is calculated for this image. Some adjustments are done in order to position the coordinates according to the plane of an image, which is different to the cartesian coordinate system. After the structure, another overlay is done, this is to show the points a, b and c on the images, and the lines linking these points. Figure 53 shows this part of the block diagram.

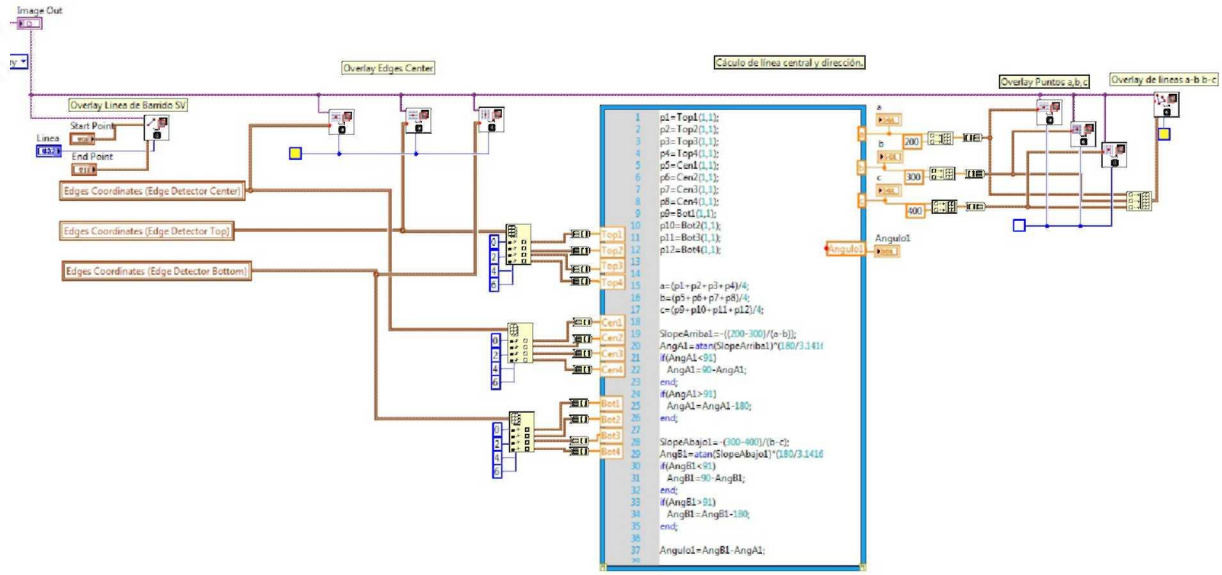


Figure 53: Overlays and calculation of parameters

The procedures in figures 50-53 are done in pairs, each for the left and right images. Having points a,b and c for each of the images and its angles, the calculation of distances is done according to equation (1). And $\theta$ as the average of the two angles. $\rho$ is calculated as the average of points in C, which are coincident with the wheels of the car. Each of the parameters is filtered to reduce its variability. Figure 54 shows this part of the block diagram.

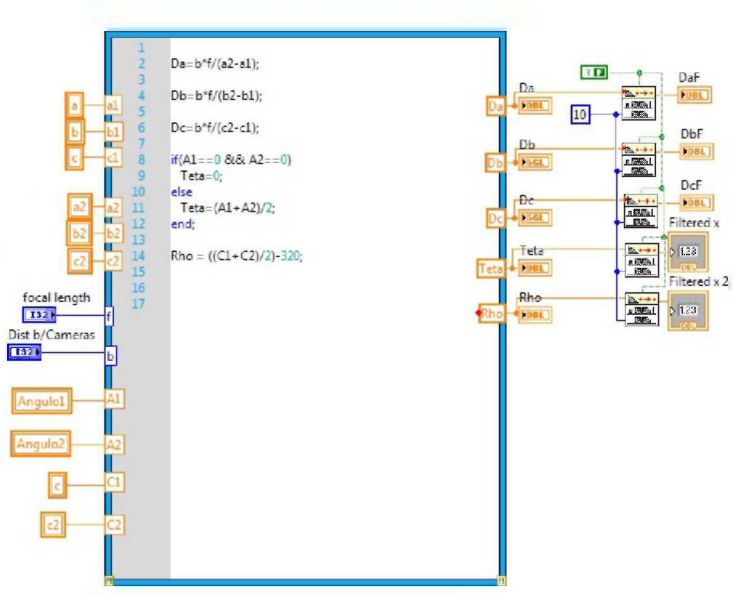


Figure 54: Distance calculation

Figures 50-54 show parts of the block diagram. In figure 55 the entire block diagram is shown.

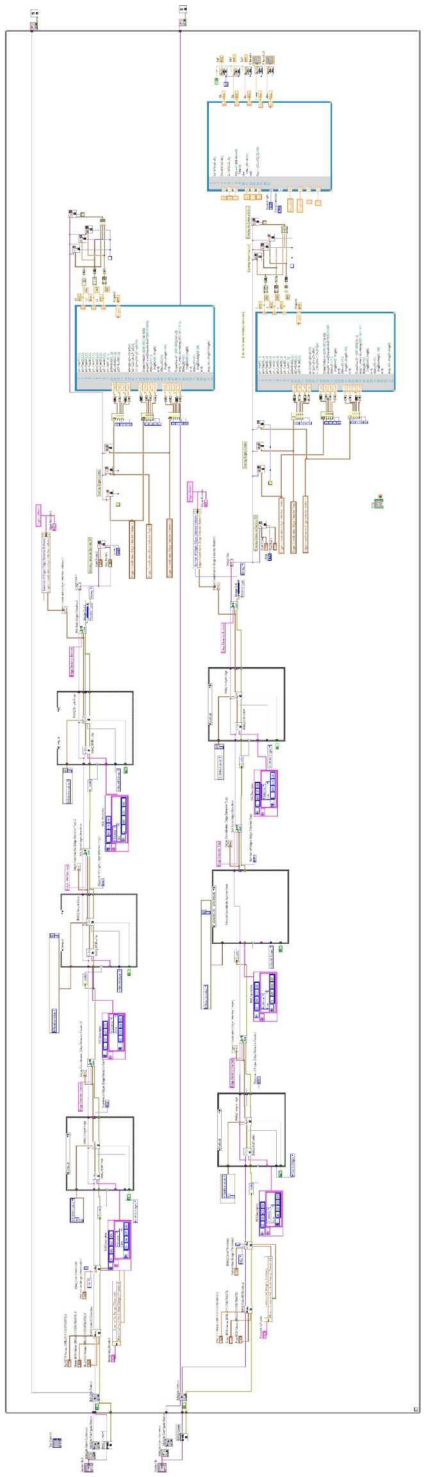

Figure 55: Block diagram.

## B.2 Fuzzy Controller

The fuzzy controller is done in a different VI(Virtual Instrument). First, the process opens a serial connection, in order to communicate to the servo controller. Then, a while loop structure is initialized to continously run; inside the while loop, the inputs are Teta and Rho, which are sent from the main VI. These variables are sent to the MISO controller (Multiple Input Single Output) with the fuzzy system, which contains the desired membership functions and defuzzification type. The output is the manipulation variable(in degrees), which is sent to a Mathscript structure, in order to convert it to a string that the servo controller recognizes. Figure 56 shows the block diagram of the fuzzy controller.

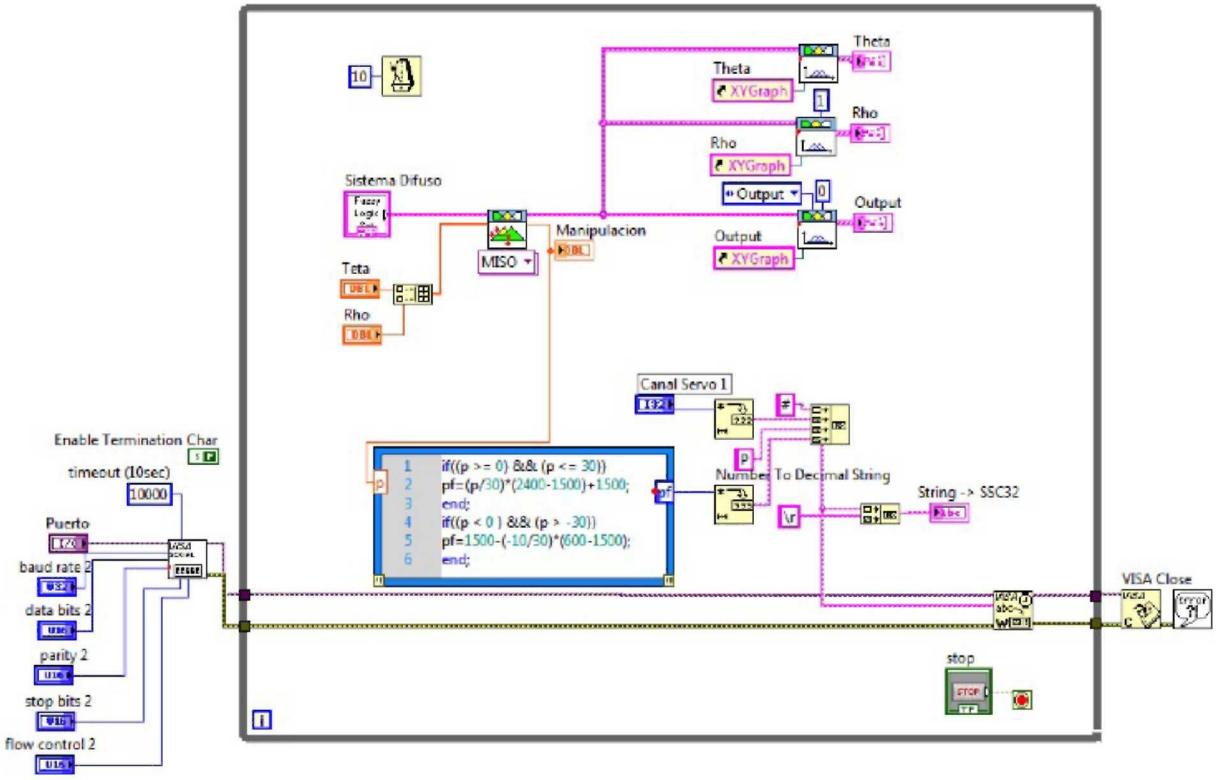


Figure 56: Block diagram of the fuzzy controller.

# C Description of Experiments

In this appendix, the explanation of how the experiments were carried out is presented. The four experiments done were: road detection, distance calculation, object detection and controllability. Table 9 shows the objective of doing each experiment.

Table 9: Experiments

| Experiment | Objective |
|---|---|
| Road Detection | To recognize the road accurately in different situations. |
| Distance Calculation | To calculate the distance between the camera and the different points accurately. |
| Object Detection | To detect objects present on the surface of the road. |
| Controllability | To know that the car maintains its position across the road. |

## C.1 Road Detection

This experiment was done in order to know that the program would recognize the road in the situations needed for this thesis. The situations are basically three: when the road is a straight line, a curve to the right and a curve to the left. Since road detection is very important for this study, it was also an important experiment.

To do this experiment, the three situations were detected. The car was put over the track and the road had to be detected correctly. The results for this experiment are shown in section 5.2, in figures 21-24.

## C.2 Distance Calculation

The experiment of distance calculation was made in order to know that the distances calculated with the program are accurate to the real distances. This experiment would ensure that the stereovision capabilities work as needed. Distance calculation is useful for the experiment explained in C.3.

In order to calculate the distances, three marks were placed at the same location of points a, b and c. The distance to those points was compared to the real distance with a meter. Figure 57 shows the left image already processed and the right image with the three white marks that correspond to the three lines checked in the left image.

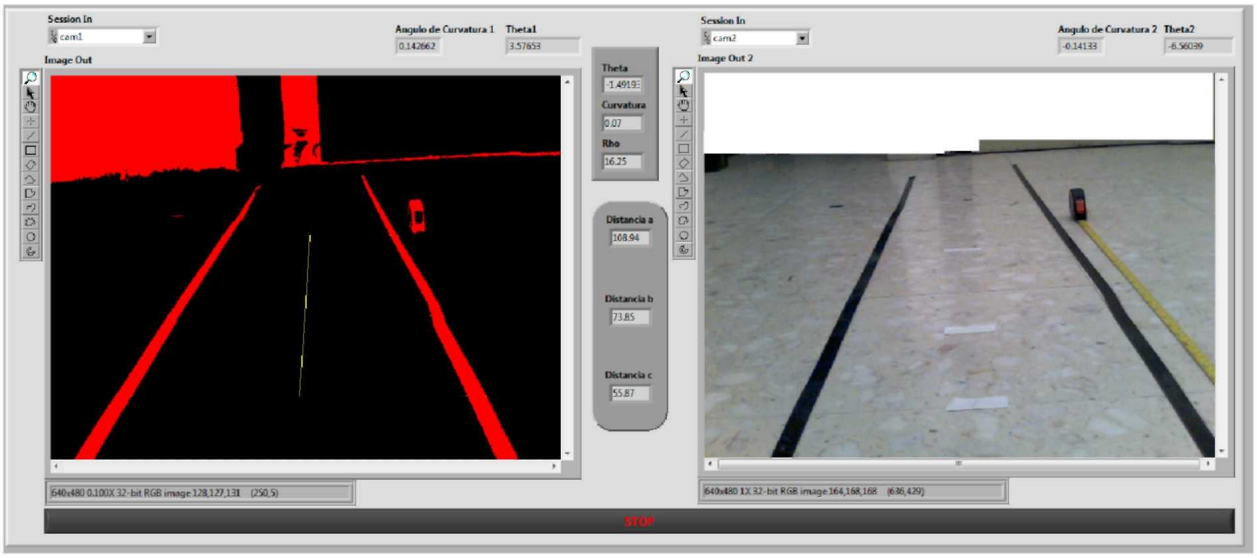


Figure 57: Experiment of distance calculation

Table 10 shows the distances calculated by the program from the camera to the points a, b and c. These distances are obtained from figure 57.

Table 10: Distances calculated

| Point | Distance Calculated |
|---|---|
| a | 111 cm. |
| b | 74.48 cm. |
| c | 56.47 cm. |

Figure 58 shows the robot with the meter in one side of the track.

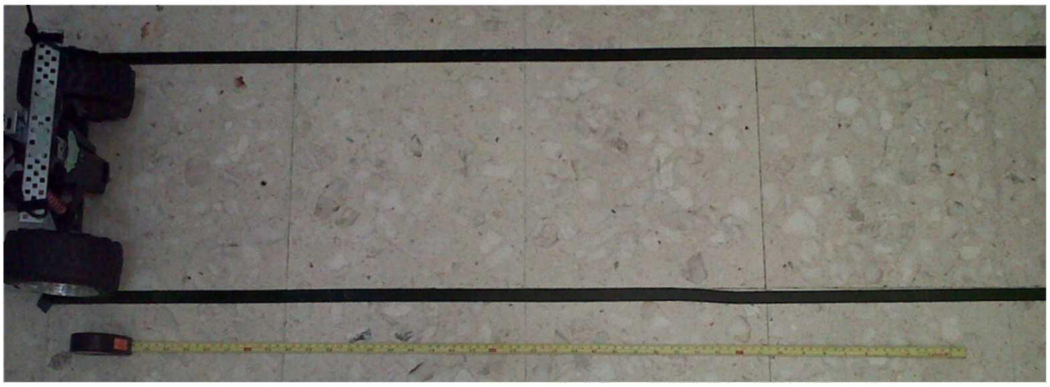

Figure 58: Experiment of distance calculation

Figure 59 shows the three white marks with its distance measured by the meter.

Figure 59: Experiment of distance calculation

The results of this experiment are shown in table 6.

## C.3 Object Detection

The experiment of object detection was carried out to make sure that the stereovision is capable of detecting the presence of obstacles on the road.

In order to carry out this experiment, objects were placed at points a, b and c. If the image processing could detect the object, then the distance to that point would change significantly and the yellow lines on the road would represent that change. The objects placed were books. Objects were also placed on the track, to analyze the obstacle detection when the object is not inside the track. As shown in figure 56, the objects were placed on the same three white marks.

The results of this experiment are presented in figures 26-30.

## C.4 Controllability

In order to know that the car is able to be driven as desired through the road, an experiment of controllability was done. The car was driven on a track and the SSE( Sum of Squared Error) was calculated for the parameter $\rho$, which is responsible for maintaining the car inside the track. Figure 60 shows the track for this experiment.

Figure 60: Track

In order to graph the results, a graph was inserted with the parameters (manipulation, rho and error) as inputs. The results of this experiment are shown in table 8 and figure 34.