VIRTUOUS: VIsion-based Road Transport-system for Unmanned Operation on Urban-like Scenarios

Miguel Angel Sotelo Vázquez

PhD Thesis

Abstract

This work describes an Intelligent Transport System implemented on an autonomous vehicle intended to perform global navigation missions in outdoor partially known environments, such as industrial or residential areas. This constitutes a first step towards the complete implementation of Intelligent Transport Systems in urban environments, which can be regarded as the long-term goal of the work. This topic is sparsely documented in the technical literature, as long as the vast majority of the already existing Intelligent Transport Systems are devoted to assisted driving of vehicles on extra urban roads and highways. Global navigation is achieved by means of a global planner, devised to compute the shortest path between the origin and some given destination, and a task manager devoted to coordinate the execution of two vision-based perception tasks for road tracking of non-structured roads, and intersection navigation, respectively, basing on GPS information. In addition, a vision-based vehicle detection task has been implemented so as to endow the global navigation system with reactive capacity. The complete system was tested on the BABIECA prototype vehicle, which was autonomously driven for hundred of kilometres accomplishing different navigation missions on a private circuit that emulates an urban quarter, at speeds up to 50 km/h. During the tests, the vehicle drove itself along crossroads and intersections performing appropriate turning manoeuvres, and demonstrated its robustness with regard to shadows, road texture, and weather and changing illumination conditions.

Contents

List of Figures

Chapter 1

Introduction

The main issue addressed in this work deals with vision based and GPS aided Intelligent Transport Systems (ITS) for autonomous global navigation in urban-like scenarios.

1.1 Motivation for Intelligent Transport Systems

The deployment of Intelligent Transport Systems in urban and extra-urban environments is a challenging topic that has focused the interest of research institutions all across the world since the mid eighties. Apart from the obvious advantages related to safety increase, such as accident rate reduction and human life savings, there are other benefits that could clearly derive from automatic driving. Thus, on one hand, vehicles keeping a short but reliable safety distance by automatic means allow to increase the capacity of roads and highways. This inexorably leads to an optimal use of infrastructures. On the other hand, a remarkable saving in combustible costs can be achieved by automatically controlling vehicles velocity so as to keep a soft acceleration profile. Likewise, automatic cooperative driving of vehicle fleets involved in the transport of heavy loads can lead to notable industrial cost reductions.

Although scenarios that allow for completely autonomous vehicles are not expected to come for at least 20 years, much of the research carried out in this area has traditionally provided remarkable procedures and techniques that find their field of application in the domain of vehicle safety enhancement. Thus, during the last five years both passive and active safety systems have received the attention of private companies in an attempt to integrate them in their commercial models. In order to avoid liability claims in the event of collisions between cars equipped with intelligent systems, manufacturers of these systems and the car companies that use them are careful to refer to them as driver aids. In this direction, adaptive cruise control (ACC) systems, which use laser beams or radar to measure the distance from the vehicle they are in to the car ahead and its speed relative to theirs, are expected to gain a global market of $2.4 billion a year by 2010 [23]. By 2006, collision avoidance will be in 17 percent of new cars in Europe, 14 percent in Asia-Pacific, and 13 percent in North America, according to Morris Kindig, president of Tier One.

1.2 ITS on highways and extraurban roads

Although the basic goal of this work is concerned with the development of an Autonomous Intelligent Transport System for urban-like environments, the techniques deployed for road tracking in this kind of scenarios are similar to those developed for road tracking in highways and structured roads, as long as they face common problems.

Nonetheless, most of the research groups currently working on ITS focus their endeavours on autonomously navigating vehicles on structured roads, i.e, marked roads. This allows to reduce the navigation problem to the localization of lane markers painted on the road surface. That's the case of some well known and prestigious systems such as RALPH [32] (Rapid Adapting Lateral Position Handler), developed on the Navlab vehicle at the Robotics Institute of the Canergie Mellon University, the impressive unmanned vehicles developed during the last decade by the research groups at the UBM [16] [27] and Daimler-Benz [18], or the GOLD system [3] [6] implemented on the ARGO autonomous vehicle at the Universita di Parma. All these systems have widely proved their validity on extensive tests carried out along thousand of kilometres of autonomous driving on structured highways and extraurban roads. The effectivity of these results on structured roads has led to the commercialization of some of these systems as driving aid products that provide warning signals upon lane depart.

On the contrary, very few research groups have undertaken the problem of autonomous vision based navigation on completely unstructured roads. Among them are the SCARF and UNSCARF systems [45] designed to extract the road shape basing on the study of homogeneous regions from a colour image. The ALVINN (Autonomous Land Vehicle In a Neural Net) [31] system is also able to follow unmarked roads after a proper training phase on the particular roads where the vehicle must navigate. Nevertheless, in spite of some promising results obtained in this field, vision based road following on unmarked roads can still be regarded nowadays as an open problem.

1.3 ITS on urban-like environments

A great interest has recently arised to design and develop Intelligent Systems for assisted driving not only on highways but in urban environments. Thus, safety enhancement becomes a very attractive point for both academic researchers and car manufacturers. According to this, the UTA project (Urban Traffic Assistant) [18] developed by the Daimler-Benz group undertook the design of an intelligent stop and go for inner-city traffic usign stereo vision, and demonstrating to recognise traffic signs, traffic lights, walking pedestrians, zebra crossings, and stop lines. Other research groups focus on partial problems, such as vision based pedestrian detection [47] [8], or intersection detection [25] [37] in order to issue warning signals to assist the human driver.

A more ambicious project was carried out at the Carnegie Mellon University aimed at recognising intersections and autonomously navigating a vehicle on them. The first objective was achieved by means of a previously trained neural network, but autonomous navigation on intersections was only solved to some extent, as the authors declare in [21]. On the other hand, another similar system can be found in [28], where a real autonomous system for Intelligent Navigation in a network of unmarked roads and intersections is designed and implemented. The vehicle is equipped with a four cameras vision system, and can be considered as the first completely autonomous vehicle capable to successfully perform some kind of global mission in an urban-like environment.

The work developed by the research group at the University of Alcalá (UAH) in the field of ITS started on 1994 with the design of a vision based algorithm for outdoor environments [36] that was implemented on an industrial fork lift truck autonomously operated on the campus of the UAH. Likewise, the research group of the *Instituto de Automática Industrial (IAI) del CSIC* has developed accurate GPS based navigation systems for autonomous guidance of commercial vehicles in urban-like scenarios under the framework of the AUTOPIA Research Programme [34]. A close cooperation between both research groups since 1999 has finally led to the development of a vision and DGPS based ITS [42] [13] for autonomous execution of global missions in a network of unstructured roads and intersections, as will be described in this paper.

The complete navigation system was implemented on BABIECA, an electric Citroën Berlingo commercial prototype as depicted in figure 1.1. The vehicle is equipped with a colour camera, a DGPS receiver, two computers, and the necessary electronic stuff to allow for automatic actuation on the steering wheel, brake and acceleration pedals. Thus, complete lateral and longitudinal automatic actuation is issued during navigation. Real tests were carried out on a private circuit emulating an urban quarter, composed of streets, intersections (crossroads), and roundabouts, located at the IAI.

Figure 1.1 BABIECA autonomous vehicle.

The work described in this paper is organised in the following sections : section II presents the complete Control Architecture for the global navigation system. In sections III and IV the vision based algorithms for lane tracking and intersection navigation are respectively described, while section V provides the presentation of a vehicle detection system for safety enhancement during navigation. In section VI the description of the lateral and longitudinal control systems is presented. Section VII presents some global results, and finally, a discussion about the whole work and concluding remarks, as well as the future work to be carried out is described in section VIII.

Chapter 2

Control Architecture

An efficient control architecture is needed so as to properly manage the information provided by the vehicle sensors (colour camera and DGPS receiver) as well as the data flow generated during navigation. The design of the control architecture considers a global system for task execution and monitoring in order to integrate the perception capabilities included in the vehicle. Likewise, a global planner is also necessary to direct and focus the behaviours of the several perception and actuation modules, basing on an a priori map of the circuit. A detailed description of these components is given below.

2.1 Environment model

A geometrical and topological description is provided to describe the real environment where the vehicle operates. The development of such a model aims at facilitating path planning. As can be derived from observation of figure 2.1, where a geometrical map of the test circuit is depicted, the geometrical representation of the operating environment resembles an urban quarter, including streets, intersections, roundabouts, and stop stations.

In the next step, the geometrical map is converted into a topological directed graph, where both the intersections and stop stations are represented by nodes, while the arcs stand for the streets that link them considering the exclusive direction of circulation. This kind of representation greatly simplifies the path planning problem, leaving the complexity of local navigation to the perception tasks. Thus, figure 2.2 shows the topological representation of the circuit map.

To give a simple example according to the map in figure 2.1, the shortest path from station 5 to station 1 implies navigating along lanes L8 and L5 until reaching intersection C2; at C2 the vehicle should turn right onto L3, and navigate along L1 and L2 to reach the final destination.

2.2 Control Architecture description

The control architecture has been divided into several clasical layers aiming at planning and executing the optimal path between the current location and the destination station as specified by the user, basing on an a priori circuit map. Global navigation is achieved by properly concatenating

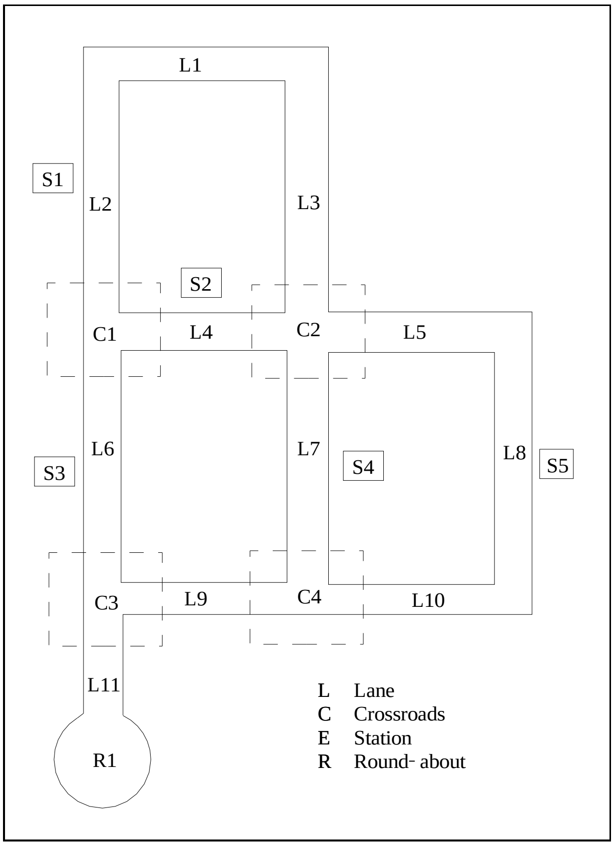


Figure 2.1 Geometrical representation of the circuit.

local perception tasks that solve vision-based navigation on streets and intersections, respectively. The same idea was suggested and successfully deployed in [21] for crosscountry navigation. The proposed architecture is depicted in figure 2.3. The basic description of the different layers included in the control scheme is provided next:

Planning layer: the global planner included in this layer computes the shortest path between the current location and the destination station, providing a recommended velocity profile for the global mission depending on whether the vehicle must navigate on a street or on an intersection. Coordination layer: the core of this layer is the task manager. It provides a link between planning and execution, by endowing the system with the capabilities of task managing and path replanning upon emergency situations or explicit user request. Navigation layer: all vision based tasks for lane tracking, intersection navigation, and vehicle detection are included in this layer. Low level: it is composed of the sensors aboard the vehicle (colour camera and DGPS receiver) together with their respective synchronised software drivers, as well as the actuator modules for the steering wheel and velocity pedals.

2.3 Global planner

According to the previously described topological model of the environment, the path planning problem can be reduced to one of traversing a mathematical graph structure composed of arcs, or

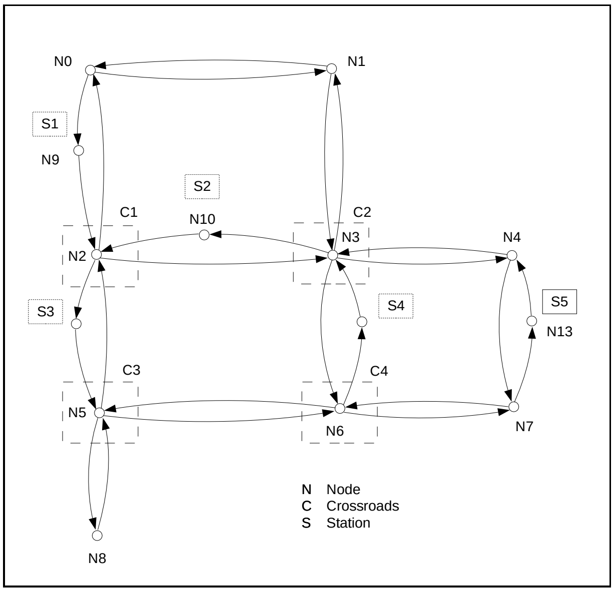


Figure 2.2 Topological representation of the circuit.

edges, and nodes, where the edges represent tracks (or streets) and the graph nodes represent the joins between tracks. To find the shortest route in this graph the popular Dijkstra algorithm [8] has been chosen. Local navigation on each section of the final route is associated to some of the following specialised vision based tasks: lane tracking and intersection navigation. Thus, edges in the graph structure are attached to the execution of lane tracking, while nodes are associated to the execution of intersection navigation. The appearance of a global plan could be something like this:

Track the lane until you reach the next intersection.
Turn right at that intersection.
Track the lane until you reach the next intersection.
Go ahead at that intersection.
Track the lane until you reach the stop station.

Likewise, an appropriate velocity profile for the different sections of the route is provided by the global planner accounting for the vehicle kinematic and dynamic constraints. Accordingly, vehicle speed will be allowed to be moderately high during lane tracking, while it will be kept low during a turn at an intersection. On the other hand, both an accelerating and a decelerating zone are considered at each intersection to gradually increase or decrease the vehicle speed depending on whether the it has completed a turn at that intersection or it is approching it, respectively. The concept is graphically depicted in figure 2.4.

The global velocity profile is computed considering the existance of these accelerating and breaking areas, yielding the typical example shown in figure 2.5. As can be observed, lane tracking is planned to be carried out at 50 km/h, quite a usual velocity in urban environments, while turns at intersections are accomplished at 5-10 km/h. On the other hand, the breaking distance d_f depends on the vehicle speed and dynamic constraints and can range from 20m to 30m in practice.

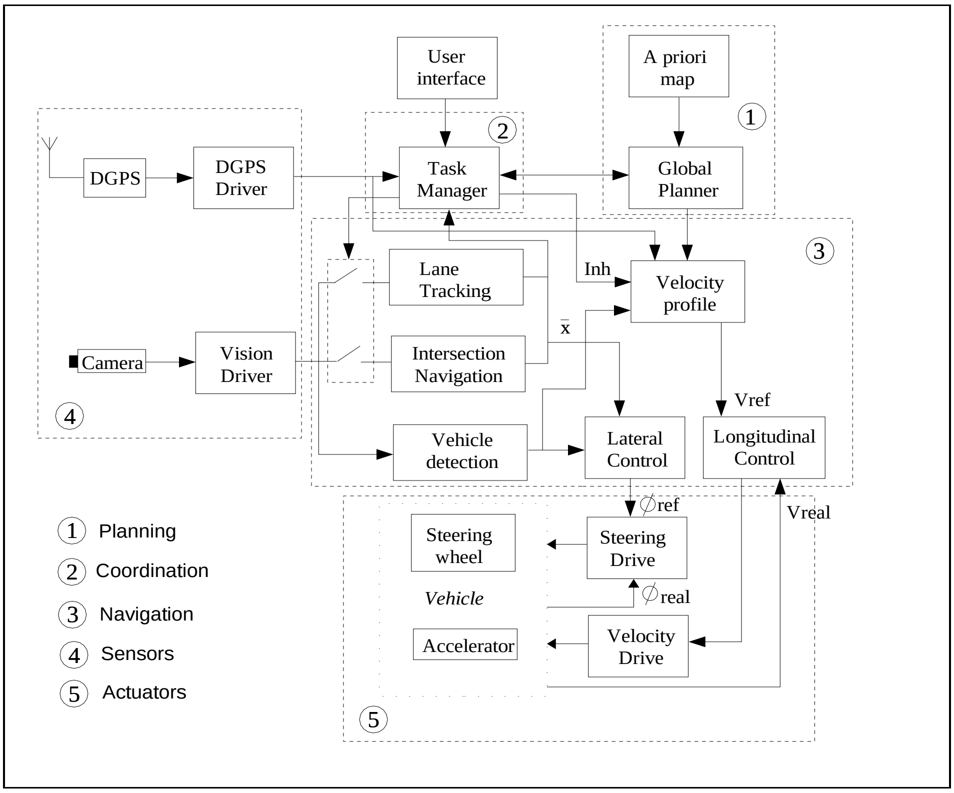


Figure 2.3 Control Architecture.

2.4 Task manager

Correct execution of a global plan involves the efficient concatenation of the local navigation tasks associated to the different sections of the route, as previously described. It becomes apparent then to design and deploy a task manager to carry out such a job. Only one task, called the active task, is executed at each time. Once the active task meets its termination condition, the task manager stops it and starts the next task according to the plan.

There are different types of termination conditions. In this work only cognitive and geometrical termination conditions have been considered. Thus, the task manager relies on an a priori map of the circuit so as to provide correct termination for lane tracking basing on the current vehicle location (as it approaches an intersection) as measured by the DGPS receiver, while navigation on intersections is terminated in a cognitive manner using the visual information contained in the scene. In conclusion, the mission of the task manager can be briefly summarised in the following points.

- Invoking the global planner at the beginning of each mission or in case of replanning upon explicit user request or emergency situations.
- Translating the global plan into a series of interlinked vision based local navigation tasks (lane tracking and intersection navigation).
- Switching between tasks after the termination conditon of the active task is met.

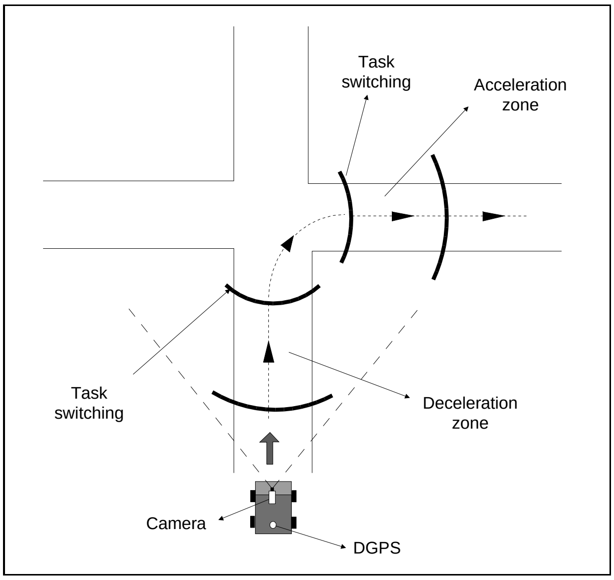


Figure 2.4 Accelerating and decelerating zones at an intersection.

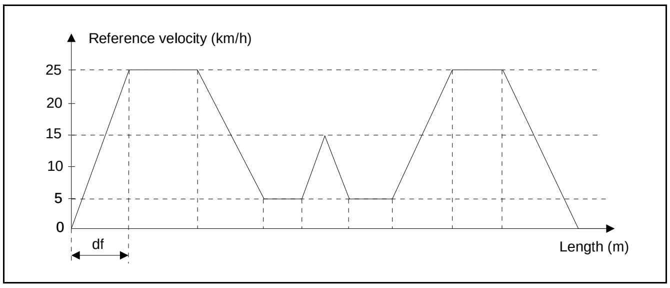


Figure 2.5 Example of velocity profile.

- Providing the vehicle with the capacity of safe emergency stop in case of failure or crash in the local navigation tasks.

Chapter 3

Lane Tracking

As described in the previous section, the main goal of this task is to correctly track the lane of any kind of road (structured or not). This includes the tracking of non structured roads, i.e, roads without lane markers painted on them.

3.1 Road model

The use of a road model eases the reconstruction of the road geometry and permits to filter the data computed during the features searching process. Among the different possibilities found in the literature, models relaying on clothoids [16] and polynomial expressions have extensively exhibited high performance in the field of road tracking. More concretely, the use of parabolic functions to model the projection of the road edges onto the image plane has been proposed and successfully tested in previous works [38]. Some of the advantages derived from the use of a second order polynomial model are described below.

- Simplicity: a second order polynomial model has only three adjustable coefficients.
- Physical plausibility: in practice, any real stretch of road can be reasonably approximated by a parabolic function in the image plane. Discontinuities in the road model are only encountered in road intersections and, particularly, in crossroads.

According to this, we've adopted the use of second order polynomial functions for both the edges and the centre of the road (the centre line will serve as a reference trajectory from which the steering angle command will be obtained), as depicted in figure 3.1.

The adjustable parameters of the several parabolic functions are continuously updated at each iteration of the algorithm using a well known least squares estimator, as will be described later. Likewise, the road width is estimated basing on the estimated road model under the slowly varying width and flat terrain assumptions. The joint use of a polynomial road model and the previously mentioned constraints allows for simple mapping between the 2D image plane and the 3D real scene using one single camera.

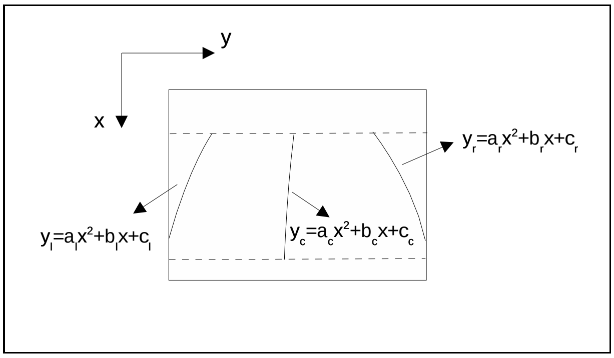


Figure 3.1 Road model.

3.2 Image Preprocessing

The original 480x512 incoming image acquired by a colour camera is in real time re-scaled to a low resolution 60x64 image, by making use of the system hardware capacities. This process aims at decreasing the whole computing time, according to the real time constraints implicit in the control of high speed vehicles for road tracking applications. It inevitably leads to a decrement in pixel resolution that must necessarily be assessed. Some of the motivations supporting this decision are cited next.

In previous works [31], it has been demonstrated that low resolution images (30x32) suffice for road tracking. The use of low resolution images allows for real time performance, strongly demanded in the road tracking problem. Likewise, the presence of other vehicles can be robustly detected using one single camera within a 20m safety distance, in spite of the resolution decrement. Nevertheless, the detection of more general obstacles, such as pedestrians, would surely require higher precision and resolution. In addition, the re-scaling process is performed in real time during image acquisition, and thus, no computing time is consumed.

3.3 Region of Interest

As discussed in [4] due to the existence of physical and continuity constraints derived from vehicle motion and road design, the analysis of the whole image can be replaced by the analysis of a specific portion of it, namely the region of interest. In this region, the probability of finding the most relevant road features is assured to be high by making use of a priori knowledge on the road shape, according to the parabolic road model proposed.

Thus, in most cases the region of interest is reduced to some portion of image surrounding the road edges estimated in the previous iteration of the algorithm. This is a valid assumption for road tracking applications heavily relying on the detection of lane markers that represent the road edges. This is not the case of the work presented in this paper, as the main goal is to autonomously navigate on completely unstructured roads (including rural paths, etc). As will be later described, colour and shape features are the key characteristics used to distinguish the road from the rest of elements in the image. This leads to a slightly different concept of region of interest where the complete road must be entirely contained in the region under analysis. On the other hand, the use of a narrow focus of attention surrounding the previous road model is strongly discarded due to the unstable behaviour exhibited by the segmentation process in practice (more detailed justification will be given in the next sections). A rectangular region of interest covering the nearest 20m ahead of the vehicle is proposed instead, as shown in figure 3.2. This restriction permits to remove non relevant elements from the image such as the sky, trees, buildings, etc, as well as to insure proper anticipation in vehicle detection, particularly in urban or industrial areas where the maximum velocity is usually under 50km/h.

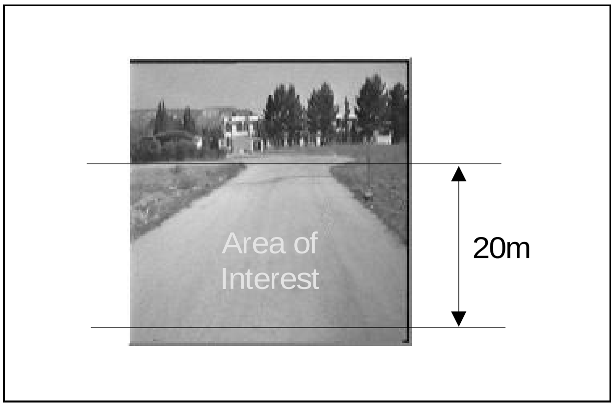


Figure 3.2 Area of Interest.

3.4 Road features

The combined use of colour and shape restrictions provides the essential information required to drive on non structured roads. Prior to the segmentation of the image, a proper selection of the most suitable colour space becomes an outstanding part of the process. On one hand, the RGB colour space has been extensively tested and used in previous road tracking applications on non structured roads [45] [12] [36]. Nevertheless, the use of the RGB colour space has some well known disadvantages, as mentioned next.

It is non intuitive and non uniform in colour separation. This means that two relatively close colours can be very separated in the RGB colour space. RGB components are slightly correlated. A colour can not be imagined from its RGB components. On the other hand, in some applications the RGB colour information is transformed into a different colour space where the luminance and crominance components of the colour are clearly separated from each other. This kind of representation benefits from the fact that the colour description model is quite oriented to human perception of colours. Additionally, in outdoor environments the change in luminance is very large due to the unpredictable and uncontrollable weather conditions, while the change in colour or crominance is not that relevant. This makes highly recommendable the use of a colour space where a clear separation between the intensity (luminance) and colour (crominance) information can be established.

The HSI (Hue, Saturation and Intensity) colour space constitutes a good example of this kind of representation, as it permits to describe colours in terms that can be intuitively understood. A human can easily recognize basic colour attributes: intensity (luminance or brightness), hue or colour, and saturation [22]. Hue represents the impression related to the predominant wavelength in the perceived colour stimulus. Saturation corresponds to the colour relative purity, and thus, non saturated colours are grey scale colours. Intensity is the amount of light in a colour. The maximum intensity is perceived as pure white, while the minimum intensity is pure black. Some of the most relevant advantages related to the use of the HSI colour space are discussed below.

It is closely related to human perception of colours. High power to discriminate colours, specially the hue component. The difference between colours can be directly quantified by using a distance measure.

Transformation from the RGB colour space to the HSI colour space can be made by means of equations 3.1 and 3.2, where V1 and V2 are intermediate variables containing the chrominance information of the colour.

$$\begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} & \frac{-2}{\sqrt{6}} & 0 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

$$H = \arctan \frac{V_2}{V_1} \quad S = \sqrt{V_1^2 + V_2^2} \tag{3.2}$$

This transformation describes a geometrical approximation to map the RGB colour cube into the HSI colour space, as depicted in figure 3.3. As can be clearly appreciated from observation of figure 3.3, colours are distributed in a cylindrical manner in the HSI colour space.

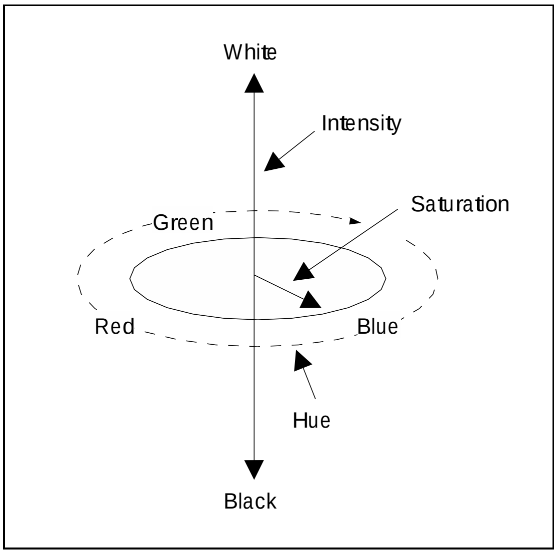


Figure 3.3 Mapping from the RGB cube to the HSI colour space.

Although the RGB colour space has been successfully used in previous works dealing with road segmentation [45] [36], the HSI colour space has exhibited superior performance in image segmentation problems as demonstrated in [10], where the concept of cromaticity saturation is the key feature for road edge detection, and [22]. According to this, we propose the use of colour features in the HSI colour space as the basis to perform the segmentation of non structured roads. A more detailed discussion supporting the use of the HSI colour space for image segmentation in outdoor applications is extensively reported in [42].

3.5 Road Segmentation

Image segmentation must be carried out by exploiting the cylindrical distribution of colour features in the HSI colour space, bearing in mind that the separation between road and no road colour characteristics is non linear. To better understand the most appropriate distance measure that should be used in the road segmentation problem consider again the decomposition of a colour vector into its three components in the HSI colour space, as illustrated in figure 3.3. According to the previous decomposition, the comparison between a pattern pixel denoted by P_p and any given pixel P_i can be directly measured in terms of intensity and chrominance distance, as depicted in figure 3.4.

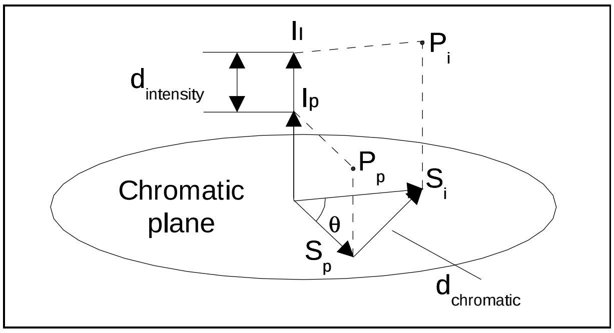


Figure 3.4 Colour comparison in HSI space.

From the analytical point of view, the difference between two colour vectors in the HSI space can be established by computing the distances both in the chromatic plane, dchromatic, and in the luminance axis, dintensity, as described in equations 3.3 and 3.4.

$$d_{intensity} = |I_p - I_i| \tag{3.3}$$

$$d_{chromatic} = \sqrt{(S_p)^2 + (S_i)^2 - 2S_pS_icos\theta} \tag{3.4}$$

with

$$\theta = \begin{cases} |H_p - H_i| & if\ |H_p - H_i| < 180° \\ 360° - |H_p - H_i| & if\ |H_p - H_i| > 180° \end{cases} \tag{3.5}$$

where H_p, H_i, S_p, S_i, I_p, and I_i represent the Hue, Saturation and Intensity of the pattern (p) and given (i) pixels, respectively. As can be readily derived from the previous equations, $d_{chromatic}$ measures the distance between two 2D colour vectors in the chromatic plane while dintensity provides the luminance difference between the pattern pixel and the pixel under consideration.

The cylindrical distribution of characteristics in the HSI colour space must be suitably exploited in order to provide an appropriate segmentation method. According to this, a cylindrical surface of separation between the road and non-road classes is proposed in an attempt to decouple chromatic changes from luminance changes, as the latter are much greater in outdoor environments despite intensity is not a determinant characteristic in the colour segmentation process. In other words, any given pixel *i* will be classified as road if the chromatic distance (dcromatic) to the colour pattern is bellow some threshold T_{chrom}, and the intensity distance ($d_{intensity}$) is lower than some T_{int}. This constraints the road pixels features in a cylinder around the pattern colour vector.

Despite hue is the most powerful colour attribute for segmentation purposes, this feature is not significant when the intensity is extremely low or extremely high. On the other hand, hue is unstable when saturation is very low, as demonstrated in [20]. According to this, pixels are divided into chromatic and achromatic as proposed in [22]. Any given pixel is considered to be achromatic if its intensity is below 10% or above 90% of the maximum normalised intensity, or if its saturation is under 10% of the maximum normalised saturation, as expressed in equation 3.6.

$$achromatic\ pixels: \quad I > 0.9 I_{max}\ or\ I < 0.1 I_{max}\ or\ S < 0.1 S_{max} \tag{3.6}$$

where I_{max} and S_{max} represent the maximum normalised intensity and saturation values, respectively. Achromatic pixels are segmented according to its intensity value alone. Obviously, non achromatic pixels are automatically categorised as chromatic. The segmentation of chromatic pixels is accomplished by applying the previously proposed cylindrical separation in the HSI colour space.

3.5.1 Adding spatial constraints

The quality of road segmentation can be strongly enhanced by adding spatial constraints according to the parabolic model used to describe the road edges. Consider the polynomial curve yc describing the trajectory of the central points of the road, projected on the image plane as depicted in figure 3.5.

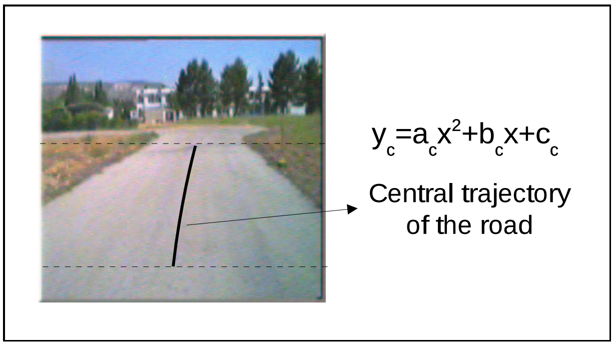


Figure 3.5 Parabolic model of the central points of the road.

In an intuitive manner, the probability that a pixel is segmented as road is high if the pixel is located close to the previous road model (as estimated in the last iteration of the algorithm), described by $y_c(t-1)$. This is particularly true for short computing time algorithms considering that in practice, due to physical constraints both in roads curvature design and in vehicle dynamics, either the road width or the temporal road model $y_c(t)$ vary gradually between two consecutive images. The last statement can be regarded as the slow varying road width assumption, widely used in previous works on road tracking [24] [32].

Incorporating spatial constraints in the segmentation stage is not a trivial process that can be accomplished in several ways. For each pixel in the image the dimension of the cylindrical surface used for segmentation is modified according to the distance from the pixel under consideration to the previously estimated road model $y_c(t-1)$, and thus, threshold values T_{chrom} and T_{int} are modulated as a function of such distance. This turns the segmentation stage into a position dependant process.

The distance d_{pi} between any given pixel i with image coordinates (x_i, y_i) and the previous road model $y_c(t-1)$ is computed on the image plane as described by equation 3.7. Graphically, the process is illustrated in figure 3.6.

$$d_{pi} = |y_i - [a_c(t-1) \cdot x_i^2 + b_c(t-1) \cdot x_i + c_c(t-1)]| \tag{3.7}$$

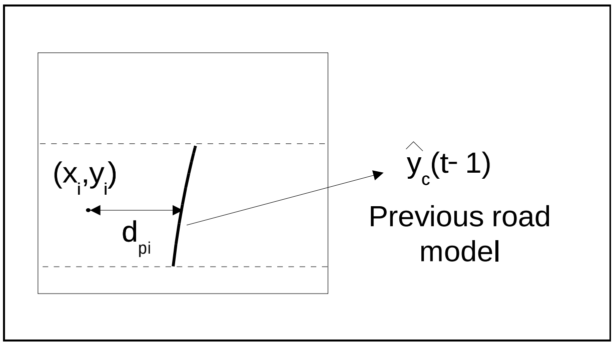


Figure 3.6 Computation of distance between pixel i and the previously estimated road model .

The real distance di between the point corresponding to the projection of pixel i on the 3D scene, and the central trajectory of the road is computed using the camera calibration parameters under the flat terrain assumption. Threshold values T_{chrom} and T_{int} are then modified for each pixel according to the distance d_i previously computed. The proposed modification is accomplished so as to provide low threshold values for pixels far away from the previous road model. Thus, for pixels clearly located out of the road trajectory, the chromatic and luminance distances to the road pattern colour features should be very small in order to effectively be segmented as part of the road. On the contrary, for pixels near the central trajectory of the previous road model those distances are admitted to be larger. Analytically, the proposal reflects an exponential variation of the threshold values T_{chrom} and T_{int} for each individual pixel i as a function of d, according to the expression in equation 3.8.

$$\begin{aligned} \Psi_c(d) &= \exp^{\frac{-K\cdot d}{\hat{W}(t-1)}} \cdot T_c(t-1) \\ \Psi_I(d) &= \exp^{\frac{-K\cdot d}{\hat{W}(t-1)}} \cdot T_I(t-1) \end{aligned} \tag{3.8}$$

where $\Psi_c(d)$ and $\Psi_I(d)$ represent the threshold values for the chromatic and luminance distances, respectively, for a pixel located at a distance d from the previous model, $T_c(t-1)$ and $T_I(t-1)$ are the maximum threshold values estimated in the previous iteration, and K is an empirically determined parameter devised to control the threshold value, particularly for pixels located in the surroundings of the road edges. In practice, K is determined so that the threshold value for pixels located at a distance from the model $d_w = \hat{W}(t-1)/2$ is 70% of the maximum threshold, yielding the numerical value depicted in equation 3.9.

$$K = -2 \cdot \ln(70/100) \tag{3.9}$$

A chromatic pixel i is classified as road if it simultaneously verifies that $d_{chrom} < \Psi_c(d)$ and $d_{int} < \Psi_I(d)$, while an achromatic pixel is segmented as road if the single condition $d_{int} < \Psi_I(d)$ is satisfied. Obviously, the initial choice of $T_c(0)$ and $T_I(0)$ becomes a critical decision whose justification is fully detailed in the next section.

On the other hand, the maximum threshold values $T_c(t)$ and $T_I(t)$ must be dynamically updated so as to adapt the segmentation process to changing colour and luminance conditions. To carry out this, the root mean squared values of the chromatic and luminance distances to the road pattern, dchrom,rms and dint,rms, are computed for each pixel classified as road. The maximum threshold values for the next iteration, $T_c(t+1)$ and $T_I(t+1)$, are calculated as described in equation 3.10 depending on $d_{chrom,rms}(t)$, $d_{int,rms}(t)$, and an exponential factor essential to guarantee the stability of the segmentation process. This leads to threshold values for the next iteration, $\Psi_c(d)_{|t+1}$ and $\Psi_I(d)_{|t+1}$, exactly equal to $d_{chrom,rms}(t)$ and $d_{int,rms}(t)$, respectively, for pixels located on the road edges ($d_{|t+1} = \hat{W}(t)/2$).

$$\begin{array}{c} T_c(t+1) = d_{chrom,rms}(t) \cdot e^{\frac{K}{2}} \\ T_I(t+1) = d_{int,rms}(t) \cdot e^{\frac{K}{2}} \end{array} \tag{3.10}$$

3.5.2 Initial Segmentation

An initial road pattern colour vector must be determined so as to provide correct features reference for the first iteration of the classification process. As can be easily imagined, no universal road features vector could be a priori considered regarding that the system should work in a long variety of different and complex scenarios, ranging from urban roads to rural paths.

An initial road pattern colour vector is computed from the information contained in the first image in a non supervised manner, assuming that the road is well within the field of view (at least one edge of the road should be visible). The colour features of pixels near the central trajectory of the initial road model are averaged to obtain the road pattern, as the probability that those pixels belong to the road is quite high. This leads to the need for determining the initial equation of the central trajectory of the road $y_c(0) = a_c(0) \cdot x^2 + b_c(0) \cdot x + c_c(0)$.

An iterative procedure is then started aiming at finding the initial road model. This model is intended to serve as the initial reference around which a set of pixels is selected to compute a candidate road pattern colour vector. Seven a priori road models are utilised for this purpose, as depicted in figure 3.7. The choice of these models has been made according to the position of the camera and its calibration parameters.

For each model, a set of M pixels is randomly selected in a surrounding of 1m around the central trajectory of the model. The HSI colour features of the M selected pixels are averaged to yield a candidate road pattern colour vector for road model i (I_{pi}, H_{pi}, S_{pi}), as indicated in equation 3.11, where a set of $M = 10$ pixels has experimentally proved sufficient to yield a representative road pattern colour vector.

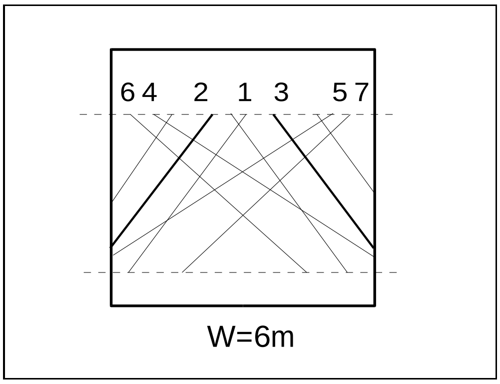


Figure 3.7 A priori road models used to determine the initial road pattern.

$$\begin{array}{c} C_i = \sum_{k=1}^{M} S_{ki} \cdot \cos(H_{ki}) \\ S_i = \sum_{k=1}^{M} S_{ki} \cdot \sin(H_{ki}) \\ H_{pi} = \arctan(S_i / C_i) \\ S_{pi} = \sqrt{C_i^2 + S_i^2} \\ I_{pi} = \frac{1}{M} \sum_{k=1}^{M} I_{ki} \end{array} \tag{3.11}$$

The maximum threshold values $T_c(0)$ and $T_I(0)$ are determined according to the quadratic distances $d_{chrom,rms-pi}$ and $d_{int,rms-pi}$ between the M selected pixels and the candidate road pattern vector for model i (H_{pi}, S_{pi}, I_{pi}), obtained as described in equations 3.12 and 3.13.

$$\begin{array}{c} d_{chrom,rms-pi} = \sqrt{\frac{1}{M} \sum_{j=1}^{M} d_{jchrom-pi}^2} \\ d_{int,rms-pi} = \sqrt{\frac{1}{M} \sum_{j=1}^{M} d_{jint-pi}^2} \end{array} \tag{3.12}$$

$$\begin{array}{c} T_c(0) = d_{chrom,rms-pi} \cdot e^{\frac{K}{2}} \\ T_I(0) = d_{int,rms-pi} \cdot e^{\frac{K}{2}} \end{array} \tag{3.13}$$

where $d_{jchrom-pi}$ and $d_{jint-pi}$ represent respectively the chromatic and intensity distances between pixel j and the candidate road pattern vector given by model i. The segmentation of the image is then carried out as described in the previous sections, using the candidate road pattern vector and an initial road width $W(0) = 6m$ (which is the standard width for a two lanes urban road). The quality of the resulting segmentation is evaluated by means of S_{0i}. Index S_{0i} is intended to validate the resulting segmentation by measuring the correlation between the initial road model i and the road segmentation obtained using the candidate road pattern colour vector computed from road model i. The process is graphically illustrated in figure 3.8.

Analytically, S_{0i} is defined as the number of road pixels in road model i matching the road pixels in the initial segmented image using such model. The final value is normalised by the total number of pixels in the image (P), as equation 3.14 shows.

$$S_{0i} = \frac{1}{P} \sum_{k=1}^{P} [1 - d_H(V_{ki}, V_{k0i})] \tag{3.14}$$

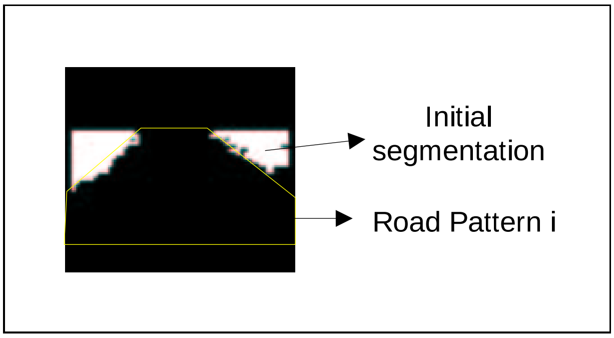


Figure 3.8 Comparison between road model i and the initial road segmentation obtained using such model.

where V_{ki} represents the binary value of segmented pixel k (0 stands for road and 1 for no road), V_{k0i} is the binary value of pixel k in the initial segmentation using road model i, and d_H is the Hamming distance. Thus, a value of S_{0i} close to unity represents a high correlation between the segmentation and model i, doubtless indicating that model i is fit to represent the real shape of the road in the image plane.

A minimum value $S_{0i,min} = 0.7$ is empirically established in order to validate the resulting segmentation. The validation process is recursively iterated until one of the N a priori road models yields a proper segmentation. If no a priori model succeeds in doing so, the width for all road models is modified in an amount of $\pm 5\%$ the nominal width. If the model road width reaches a modification of $\pm 20\%$ the nominal width without producing a valid segmentation, the iterative process is restarted from the standard value $W = 6m$. The complete algorithm to perform the initial segmentation is depicted in figure 3.9.

Experimental results have successfully proved the ability of the proposed strategy to carry out the initial segmentation, yielding reasonably neat binary images for arbitrary initial vehicle orientations, while keeping the processing time under 1ms for most of the experiments conducted on a real scenario, as depicted in figure 3.10.

To enhance the quality of the segmentation process the resulting binary image is reinforced by a morphological opening operation followed by the removal of small white blobs corresponding to segmentation noise. The benefits derived from these operations can be graphically appreciated in figure 3.11.

3.6 Handling shadows and brightness

Shadows and brightness on the road are admittedly the greatest difficulty in vision based systems operating in outdoor environments [3]. The problem affects the detection of both lane markings and road edges in general, becoming specially dangerous at some particular hours of the day when the sun directly shines onto the image plane, deriving in situations where loss of tracking occurs.

In order to deal with this problem, some authors propose to improve the dynamic range of visual cameras [4] so as to tackle strong luminance changes, when entering or exiting tunnels for instance, or to enhance the sensitiveness of cameras to the blue component of colours. A different approach undertakes alone the problem of shadows by attenuating their effects using an appropri-

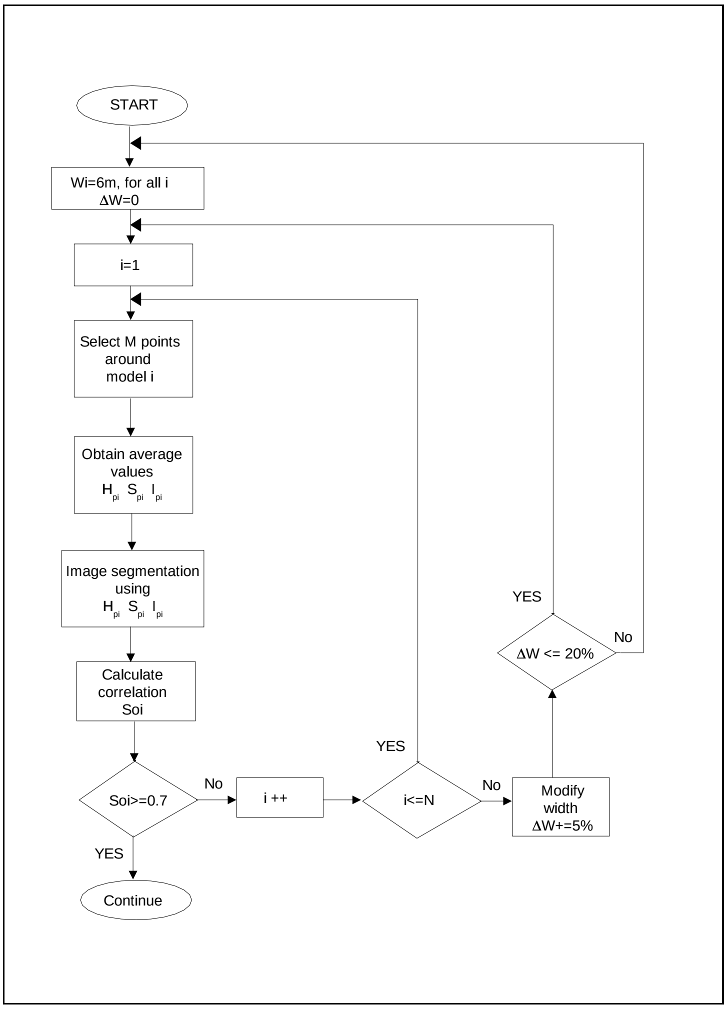


Figure 3.9 Algorithm for initial segmentation.

ate software pre-processing technique, relaying on physical properties of shaded road pixels. On one hand, shaded road pixels exhibit lower intensity values than their neighbours corresponding to non-shaded road pixels. On the other hand, the normalised blue component is generally predominant over the normalised red and green components, as discussed in [31]. According to this, the addition of both the blue and the normalised blue components is then exploited to attenuate the effect of shadows, to some extent. This approach is intended to yield a low resolution grey scale image that serves as input for a neural network that directly obtains the vehicle turning angle, assuming that no colour information is used thereinafter.

In the current work, a slightly different strategy is formulated so as to enhance the resulting segmentation against the effects of both shadows and brightness. This is realised by accounting for colour properties of pixels located within the road edges, as estimated in the previous iteration of the algorithm, in an attempt to incorporate spatial constraints in the shadows and brightness attenuation process. Thus, colour features of pixels located within the limits of the road, but classified as non-road after the segmentation process, are considered for brightness and shadows attenuation.

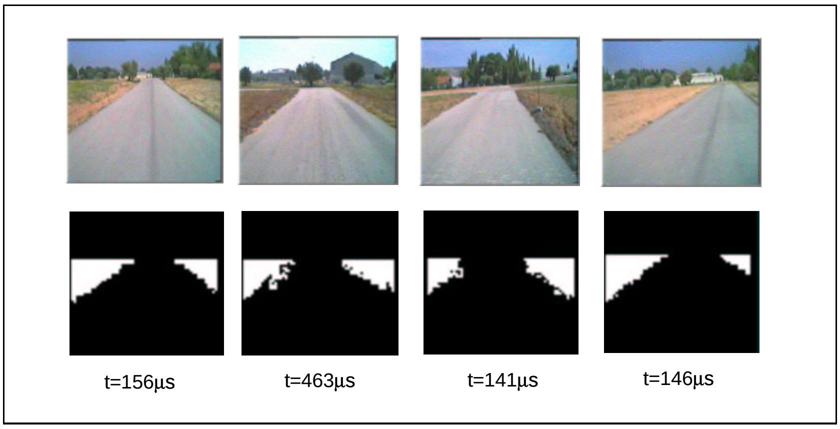


Figure 3.10 Examples of initial segmentations.

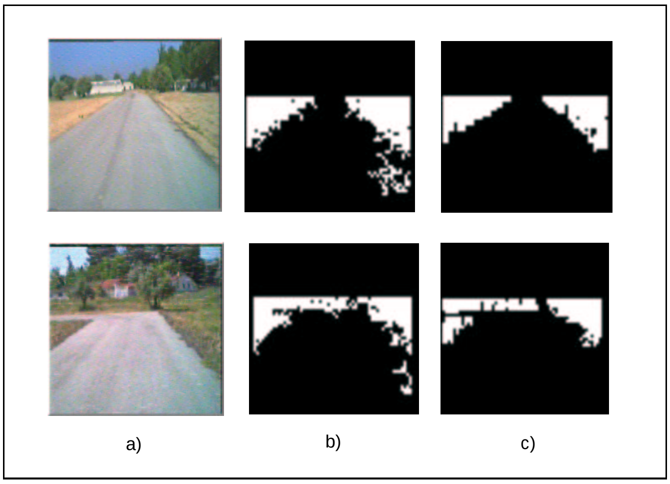


Figure 3.11 Segmentation results after morphological post-processing.

In a first step, shaded pixels should simultaneously exhibit an intensity value lower than the average intensity of road pixels, while presenting a predominant normalised blue component. Those pixels complying with the conditions previously described, and analytically expressed in equation 3.15, are assumed to belong to a shadow on the pavement and will be consequently reclassified as road pixels.

$$\begin{gathered} b \geq \frac{1}{3} \\ I \leq I_{road,avg} - 2 \cdot \sigma_{road} \end{gathered} \tag{3.15}$$

where b stands for the normalised blue component; $I_{road,avg}$ represents the average intensity value of all road pixels, and σ_{road} is the standard deviation of the intensity distribution of road pixels. This technique permits to enhance the road segmentation in presence of shadows, and remarkably contributes to improve the robustness of the colour adaptation process, particularly in stretches of road largely covered by shadows. To graphically illustrate the benefits derived from this operation, figure 3.12 shows an example of road segmentation in presence of strong shadows. As can be appreciated, the road edges are neatly distinguished after the attenuation of shadows.

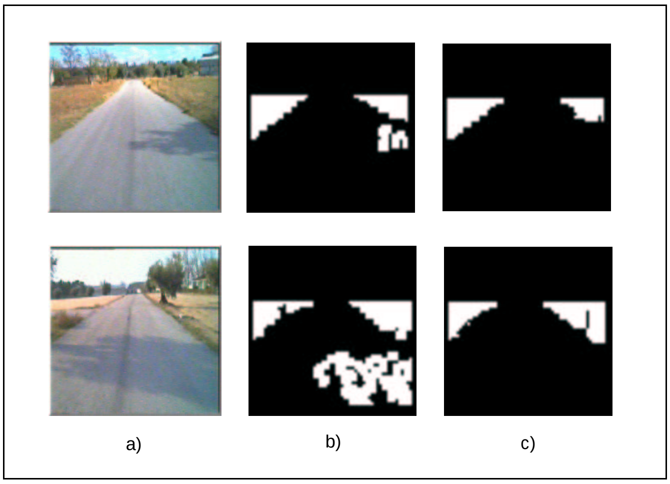


Figure 3.12 Attenuation of shadows. a) Original shaded images. b) Segmentation without attenuation of shadows. c) Segmentation after attenuation of shadows.

Analogously, a brightness attenuation technique has been devised. In this case, pixels initially classified as non-road but located within the road edges and exhibiting higher intensity values than the average road pixels, are assumed to correspond to brightness on the pavement caused by the sun, and consequently will be re-classified as road-pixels. Analytically the condition is formulated in equation 3.16.

$$I \geq I_{road,avg} + 2 \cdot \sigma_{road} \tag{3.16}$$

where $I_{road,avg}$ and σ_{road} are the same variables considered in equation 3.15. After applying the condition established by equation 3.16, white blobs due to brightness are removed from the segmentation as depicted in figure 3.13. The improvement achieved by attenuating both brightness and shadows as described permits to handle real images in real and complex situations with an extraordinary high performance, becoming an outstanding point of this work.

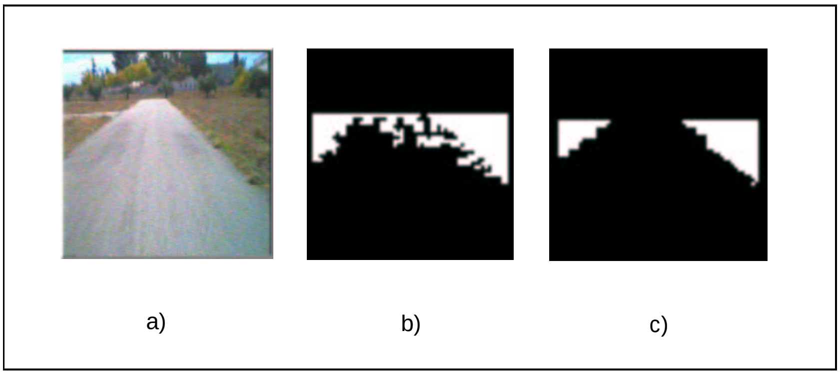


Figure 3.13 Brightness attenuation. a) Original image with brightness on the pavement. b) Segmentation without brightness attenuation. c) Segmentation after brightness attenuation.

3.7 Estimation of road edges and width

The estimation of the central trajectory of the road and its edges is carried out basing on parabolic functions, as described in section 3.1. These polynomial functions are the basis to obtain the lateral and orientation error of the vehicle with respect to the centre of the lane.

3.7.1 Initial conditions

As previously mentioned, the initial segmentation was derived basing on basic pattern i, whose width and edges are regarded as the initial values for the estimation process, i.e, $\hat{W}(t=0)=W_i$, $\hat{y}_l(0)=y_{li}$, $\hat{y}_c(0)=y_{ci}$, and $\hat{y}_r(0)=y_{ri}$, where $\hat{W}(t=0)$ represents the initial estimation of road width, and Wi is the width of basic pattern i. On the other hand, $\hat{y}_l(0), \hat{y}_c(0), \hat{y}_r(0)$ are the initial estimation for the left edge, right edge, and central trajectory of the road, respectively, while y_{li}, y_{ci}, y_{ri} stand for the left edge, right edge, and central trajectory of basic pattern i.

3.7.2 Estimation of the central trajectory of the road

The central trajectory of the road at current time instant, $\hat{y}_c(t)$, is estimated basing on the segmented low resolution image and the previously estimated road trajectory, $\hat{y}_c(t-1)$. Temporal correlation among measures obtained at different instants of time is considered, as well as the number of data used to carry out the estimation, so as to enhance the road estimation process. Accordingly, a weighted-recursive least squares estimator with exponential decay is utilised for this purpose as initially proposed in [38]. The analytical formulation of this filtering technique is quite similar to that of Kalman filter.

Data measurement

The objective of this first stage is to extract a number of candidate points associated to the central trajectory of the road at time instant t. For each line k in the region of interest the maximum road width is determined basing on the segmented image, as depicted in figure 3.14. The middle point of each maximal road width line k is considered as candidate, and its coordinates $(y_{t,k}, x_{t,k})$ are validated if the road width of line k is greater than some threshold. This intends to account for noise rejection.

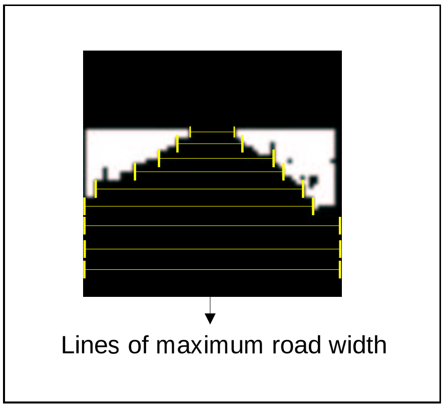


Figure 3.14 Maximum road width for each line in the region of interest.

Data association and validation

In order to provide the algorithm with noise rejection capacity, only candidate pixels whose distance to the previous estimation $\hat{y}_c(t-1)$ is under some threshold V are validated and associated to the current measure of the central trajectory of the road. This permits to establish a validation area around the previous road model as depicted in figure 3.15. All measures residing out of the validation area are discarded and regarded as invalid measures.

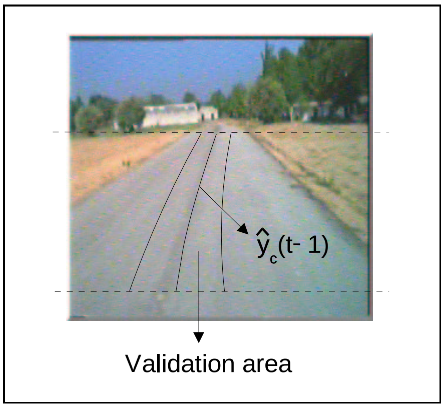


Figure 3.15 Validation area for measures associated to the central trajectory of the road.

Figure 3.16 depicts the points measured for a sequence of four images in a real scenario, using the proposed validation area. Obviously, measures corresponding to lines where the maximum road width coincides with the image width (i.e, lines where only road pixels are perceived), are also disregarded as valid measures as long as those lines don't provide any information about the road edges. The effect of these false measures can be graphically appreciated as straight segments in the bottom-middle part of the example images illustrated in figure 3.16.

Figure 3.16 Data associated to the central trajectory of the road in a sequence of images.

Road model update

Measures validated in the last stage constitute the starting point for updating the parabolic road model. As previously mentioned, a weighted-recursive least squares estimator with exponential decay is proposed to perform the estimation of the central trajectory of the road. In spite of being a well-known and well-documented theory, we repeat the basic equations in this paper for completeness reasons. Thus, the estimation is realised in three steps as described below.

a)Update prediction.

$$\hat{z}(t) = \phi^T(t) \cdot \theta(t-1) \tag{3.17}$$

b)Update state covariance estimate.

$$P(t) = \frac{1}{\lambda}[P(t-1) - [P(t-1)\phi(t) \cdot (\lambda I + \phi^T(t)P(t-1)\phi(t))^{-1} \cdot \phi^T(t)P^T(t-1)]] \tag{3.18}$$

c)Update state estimate.

$$\theta(t) = \theta(t-1) + G(t) \cdot [z(t) - \hat{z}(t)] \tag{3.19}$$

where

$$G(t) = P(t-1)\phi(t) \cdot (\lambda I + \phi^T(t)P(t-1)\phi(t))^{-1}$$
$$z(t) = \begin{bmatrix} y_{t,1} \\ y_{t,2} \\ \cdots \\ y_{t,N_t} \end{bmatrix} \quad \phi^T(t) = \begin{bmatrix} 1 & x_{t,1} & x_{t,1}^2 \\ 1 & x_{t,2} & x_{t,2}^2 \\ \cdots & \cdots & \cdots \\ 1 & x_{t,N_t} & x_{t,N_t}^2 \end{bmatrix} \quad \theta(t-1) = \begin{bmatrix} c \\ b \\ a \end{bmatrix}$$

where $\theta(t-1)$ represents the state estimation at the previous iteration of the algorithm, i.e. at time instant $t - \Delta T_v$ (ΔT_v is the sampling period of the road tracking algorithm), λ is a scalar value that can vary in the range $0 \leq \lambda \leq 1$, and $P(t-1)$ stands for its covariance. To achieve a proper trade-off between robustness and transient response, λ has been experimentally set to 0.7, exhibiting an adequate performance in real tests. To illustrate the estimation process, figure 3.17 shows the results achieved upon a sequence of real images, using the proposed formulation.

3.7.3 Road edges estimation

The estimation of road edges is realised using the same filtering technique described in the previous section. Measures for the left and right road edges are validated and enhanced basing on three fundamental points: the estimation of the central trajectory of the road at current time t, the estimation of road width at time $t-1$, and the slowly varying road width assumption. Thus, a validation area is also established for the left and right measures, as depicted in figure 3.18. The locations of the left and right validation areas are based on the central trajectory of the road estimated at time t, and the estimated width at time $t-1$. The left edge validation area is placed on the left, $\hat{W}(t-1)/2$ meters away from the central trajectory of the road $\hat{y}_c(t)$, while the right edge validation area is obviously located to the right, at $\hat{W}(t-1)/2$ meters from $\hat{y}_c(t)$.

Consequently, for each line in the area of interest, the closest measures to the middle of the left edge validation area, defined by $\hat{y}_c(t) - \hat{W}(t-1)/2$, and right edge validation area, defined by

Figure 3.17 Estimation of the central trajectory of the road in a sequence of real images.

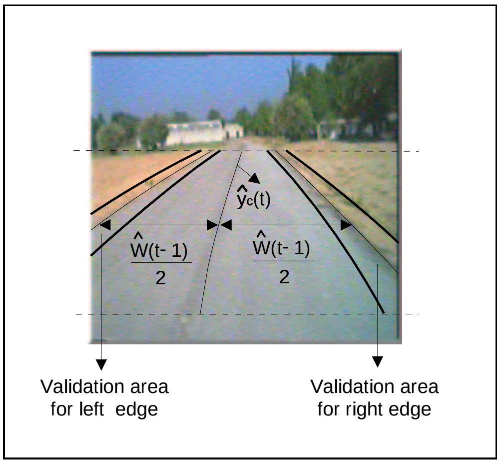


Figure 3.18 Validation areas for edge measures.

$\hat{y}_c(t) + \hat{W}(t-1)/2$, are considered and validated if the distance to the respective edge reference is below $V = 1m$. The estimations of the left and right edges are independently carried out basing on the validated measures for each edge (N_{lt} points for the left edge and N_{rt} for the right edge), while having different covariance matrices ($P_l(t)$ for the left edge, $P_r(t)$ for the right edge). Upon the conclusion of the estimation process, vector $\theta_l(t) = (a_l, b_l, c_l)^T$ determines the coefficients of the parabolic polynomial that approximates the left road edge, while $\theta_r(t) = (a_r, b_r, c_r)^T$ determines the coefficients for the right edge. A complete example of image segmentation, edge points extraction, and road edges estimation is depicted in figure 3.19.

3.7.4 Road width estimation

The road width is an essential parameter in the complete road tracking scheme. Its estimation is realised on the basis of the previously mentioned slowly varying road width assumption. An individual road width measure w_i is obtained for each line in the region of interest, by computing the

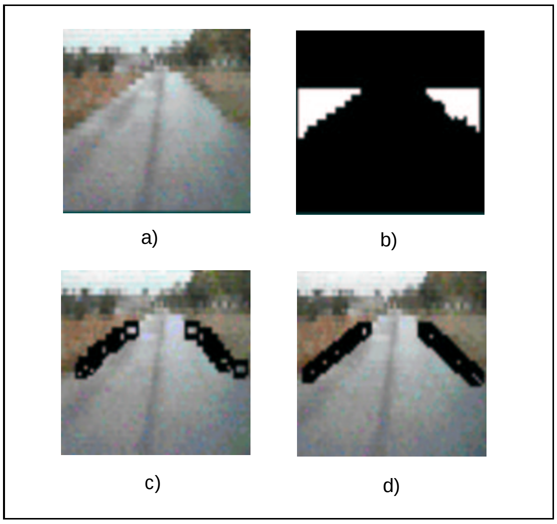


Figure 3.19 A complete example. a) Real image. b) Segmentation of the region of interest. c) Validated edge points. d) Road edges estimation.

difference between the left and right edges ($\hat{y}_l(t)_{|x=x_i}$ and $\hat{y}_r(t)_{|x=x_i}$), respectively) as expressed in equation 3.20.

$$w_i = \hat{y}_r(t)_{|x=x_i} - \hat{y}_l(t)_{|x=x_i} \tag{3.20}$$

Measures w_i obtained in the image plane are properly corrected, using the calibration parameters, to yield real measures in the 3D scene. The average road width measure at time t, $W(t)$, is computed using the individual measures for each line, normalised by the number of valid measures in the region of interest N_{rdi}, as in equation 3,21.

$$W(t) = \frac{1}{N_{rdi}} \sum_{i=1}^{N_{rdi}} w_{r.i} \tag{3.21}$$

The slowly varying road width assumption is incorporated using a recursive least squares based estimator, similar to those employed for the estimation of road edges. This permits to issue a smooth estimation of the road width. Its analytical formulation is presented below.

$$\hat{m}(t) = \hat{W}(t-1) \tag{3.22}$$

$$P_w(t) = \frac{1}{\lambda_w}(P_w(t-1) - K_w(t)P^T(t-1)) \tag{3.23}$$

$$\hat{W}(t) = \hat{W}(t-1) + K_w(t)(W(t) - \hat{m}(t)) \tag{3.24}$$

with

$$K_w(t) = P_w(t-1)/(\lambda_w + P_w(t-1)) \tag{3.25}$$

where $\hat{m}(t)$ represents the prediction of the current state, and $P_w(t)$ stands for covariance of the estimated width. In practice, proper results have been achieved using $\lambda_w = 0.8$. To graphically illustrate the whole process, figure 3.20 depicts a sequence of road images showing the estimation of the central trajectory and edges of the road, as well as the estimated road width.

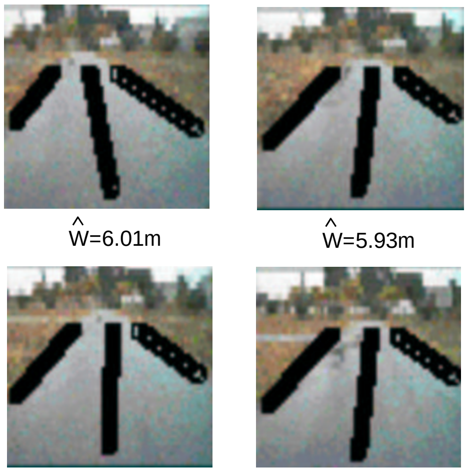


Figure 3.20 Estimation of road edges and width in a sequence of images.

3.8 Road colour features update

After completing the road edges and width estimation process, the HSI colour features of the road pattern are consequently updated so as to account for changes in road appearance and illumination. Intuitively, pixels close to the central trajectory of the road present colour features that highly represent the road colour pattern in general. Accordingly, a set of $N_p = 8$ pixels in a region of $1m$ surrounding the central estimation of the road, $\hat{y}_c(t)$, is randomly chosen as depicted in figure 3.21. Obviously, the selected pixels are only validated if they have been segmented as road pixels at the current iteration.

The HSI colour features of the road pattern are properly averaged basing on the individual HSI characteristics of the selected pixels, as shown in equations 3.26 and 3.27.

$$\begin{aligned} C_M &= \sum_{k=1}^{N_p} S_k \cdot \cos(H_k) \\ S_M &= \sum_{k=1}^{N_p} S_k \cdot sen(H_k) \\ H_p &= \arctan(S_M / C_M) \\ S_p &= \sqrt{C_M^2 + S_M^2} \end{aligned} \tag{3.26}$$

$$I_p = \frac{1}{N_p} \sum_{k=1}^{N_p} I_k \tag{3.27}$$

The adaptation process described in this section proves to be crucial in practice to keep the segmentation algorithm under stable performance upon illumination changing conditions and colour

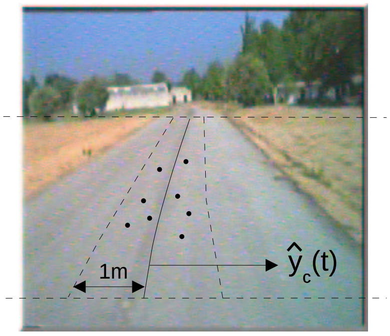


Figure 3.21 Random pixel selection for HSI road pattern update.

varying asphalt. The complete road tracking scheme is graphically summarised in the flow diagram depicted in figure 3.22.

3.9 Discussion

The global objective of this section is to put the road tracking algorithm under test in complex and varied real circumstances. Thus, we propose to analyse the system performance on different non structured roads in strongly changing weather and illumination conditions as described below.

3.9.1 Non structured roads

In a first set of trials, the road tracking algorithm is evaluated on a private circuit whose structure resembles the typical layout of an industrial area. Figure 3.23 depicts an example of road segmentation obtained on this scenario. As appreciated from observation of figure 3.23, the road edges can be neatly distinguished in the segmented image, allowing a clear and robust estimation in real experiments.

In a second trial, similar experiments were conducted on rural roads. As observed in figure 3.24, a central lane marker is clearly painted on the asphalt, while no lane markers are present at all near the road edges. In spite of these structural conditions, the segmentations depicted in figure 3.24 strongly support the ability of the road tracking algorithm to successfully perform in this kind of scenario.

The system has also been evaluated on a University Campus, under typical urban conditions such as zebra crossings, parked vehicles, etc, using a set of recorded images. Thus, figure 3.25 depicts two different and representative situations on urban driving. On one hand, figure 3.25a shows the segmentation and road edges estimation in presence of a zebra crossing. As can be observed, the segmentation algorithm provides an efficient filtering for this kind of perturbation, while a robust estimation of the road edges is preserved. On the other hand, the results obtained in presence of other vehicles are illustrated in figure 3.25b. Correct segmentation and edges estimation are

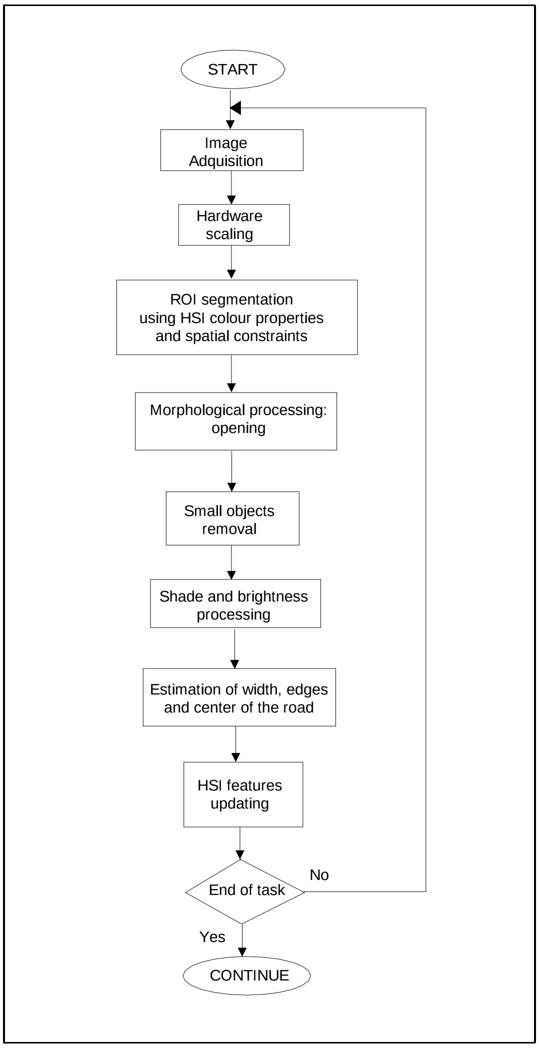


Figure 3.22 Road tracking flow diagram.

also achieved in this case.

In a final trial, the road tracking scheme is put under evaluation on roads without asphalt. Accordingly, the edges of a narrow rural path are correctly estimated basing on the road segmentation, as depicted in figure 3.26.

3.9.2 Robustness against environmental and weather conditions

In this section, the performance of the road tracking algorithm is evaluated under different environmental and weather conditions, so as to verify the validity and generality of the segmentation and updating scheme. All tests were conducted on a private circuit, previously mentioned in the last section.

Sunny conditions

The excessive amount of luminance becomes a major problem when dealing with road images

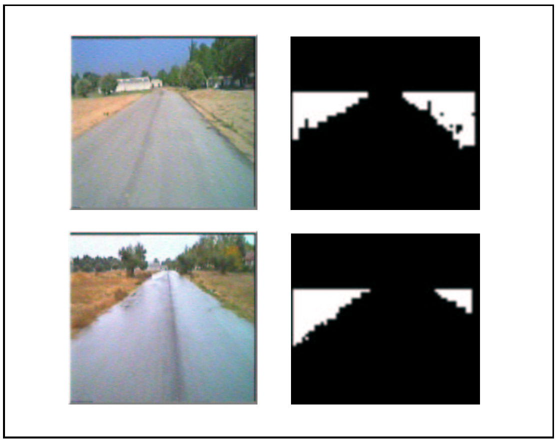


Figure 3.23 Road segmentation obtained on a private circuit.

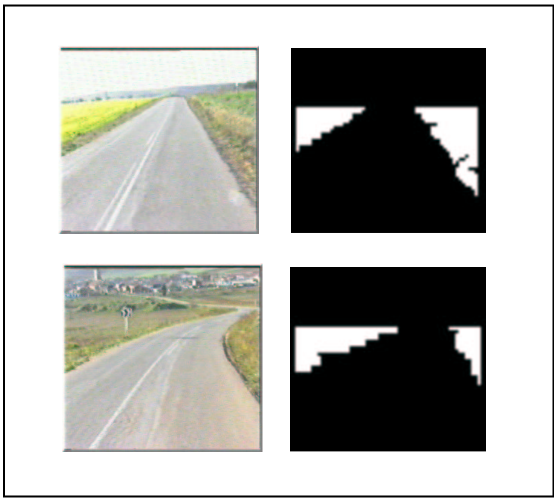


Figure 3.24 Segmentations obtained on rural roads.

in a sunny day. All pixels in the image tend to have similar intensity values, and thus, colour differences in the HSI chromatic plane become crucial for segmentation purposes. In spite of having achieved correct performance in real experiments under sunny conditions in general terms, a few remarks must be pointed out.

There exist limitations in the schedule of applicability due to direct incidence of sunrays onto the camera lens, just after sunrise and before sunset in strongly sunny days. Accordingly, autonomous navigation becomes dangerous and non-advisable under these conditions that obviously depend on the latitude and the season of the year.

The presence of strong and large shadows on the road especially after sunrise may complicate the segmentation process to the extent of making recommendable to decrease vehicle velocity, in order to avoid driving oscillations. Nevertheless, transitions from shaded to sunny areas as well as navigation on completely shaded zones are adequately managed by the segmentation and updating algorithm out of the critical hours before mentioned.

Cloudy and rainy conditions

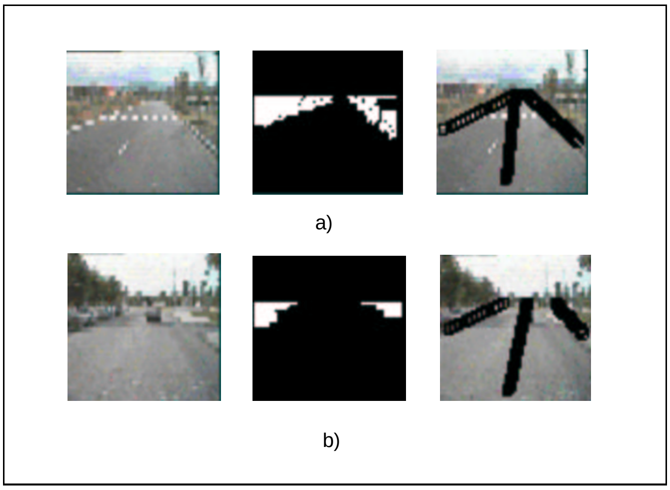


Figure 3.25 Segmentation and road edges estimation on urban areas in presence of a) a zebra crossing, b) other vehicles.

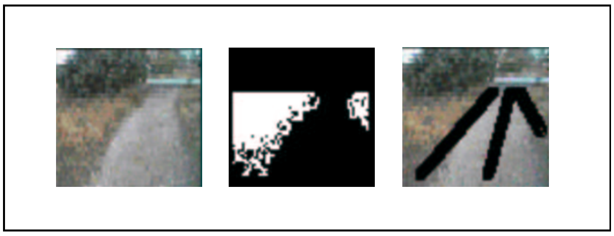


Figure 3.26 Segmentation and road edges estimation on a rural path without asphalt.

The amount of intensity in the image strongly decreases on cloudy and rainy days. Paradoxically, this circumstance eases the discrimination process between road and non-road pixels, as colour differences become larger. Figure 3.27 depicts a typical example of image segmentation and road edges estimation on cloudy conditions. As can be observed, the high quality of the road segmentation shown in figure 3.27 doubtless support the previous discussion.

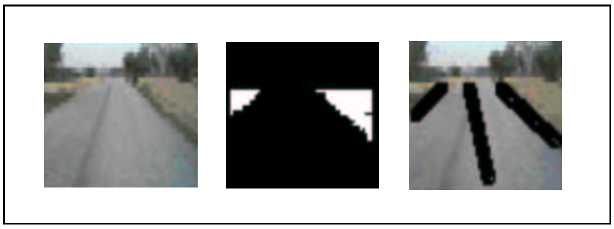


Figure 3.27 Segmentation and edges estimation on cloudy conditions.

Likewise, the road takes quite a characteristic colour on rainy days because of the water on it. This also contributes to a better and easier separation between road and non-road pixels. Even puddles on the road are correctly segmented, as graphically demonstrated in figure 3.28, where both the image segmentation and road edges estimation are illustrated for a typical rainy scene.

However, the most dangerous situation takes place after the rain stops, as the asphalt gets dry in a non-homogeneous manner. This situation yields to the appearance of dark spots on the road

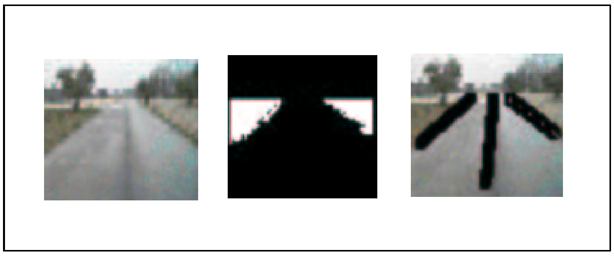


Figure 3.28 Segmentation and edges estimation on rainy conditions.

due to wet areas, as depicted in figure 3.29. Fortunately, the segmentation process (including the small blobs removal stage) properly manages these circumstances allowing to obtain high quality segmentations, as shown in figure 3.29.

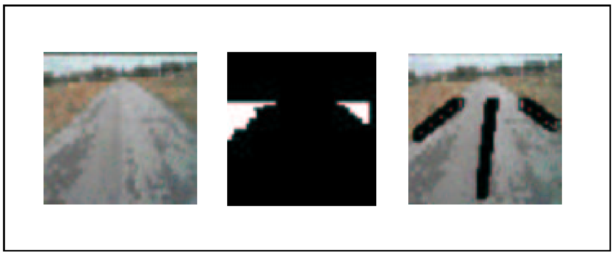


Figure 3.29 Segmentation and road edges estimation on post-rainy conditions.

Foggy days

In general, autonomous navigation is not advisable on foggy days, even for humans. In spite of this, incredibly proper segmentations can be obtained under non-heavy foggy conditions, as depicted in figure 3.30.

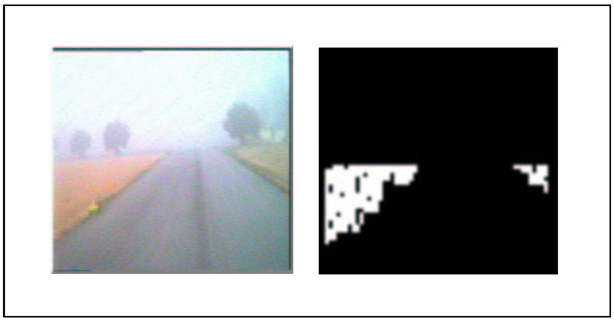


Figure 3.30 Road segmentation on non-heavy foggy conditions.

3.10 Conclusions

The previous discussion lets us state that the road segmentation algorithm based on the HSI colour space and 2D-spatial constraints proves successfully to provide an accurate and robust estimation for the edges and width of non-structured roads, i.e., roads without lane markers. The practical results discussed above also support the validity of the method for different environmental and weather conditions, as demonstrated. On the contrary, there exist some limitations in the use of the road tracking algorithm just after sunrise and before sunset in very sunny days, primarily due to direct incidence of sunrays onto the camera plane.

The most remarkable feature of the road tracking scheme described in this work is its ability to correctly deal with non-structured roads, as this kind of scenario hasn't received much attention from the international scientific community during the last years. This stems from the fact that many of the greatest automotive companies across the globe are currently focusing their financial resources on developing vision based commercial products for assisted driving on structured roads. That's the case of lane depart warning systems, prediction of curvature radius, night vision enhancement systems, traffic signal detection and recognition, etc. Such a huge economical support made research groups shift from non-structured roads (difficult to deal with and with no commercial applicability in the short and mid terms) to structured roads (such as highways, offering an enormous market for industrial companies) where the lanes are clearly determined by white or yellow lane markers (and thus, allowing simpler and faster algorithms to be deployed).

In order to establish a comparative discussion, two previous research works are remarkably over the rest in the domain of vision based autonomous navigation on non-structured roads.

On one hand, the SCARF-UNSCARF system [45] provided one of the first road tracking algorithms for autonomous driving on non-structured roads, physically tested and demonstrated on a real vehicle. The SCARF system realised a supervised classification, using colour pre-computing, obviously failing upon strongly varying lighting conditions. UNSCARF solved these problems by applying an unsupervised iterative clustering technique based on statistical models for the road and non-road classes. RGB colour features of individual pixels, as well as their row and column positions are used for this purpose, yielding a five components classifier. Two major drawbacks can be pointed out under this approach: poor precision is achieved in determining the road edges, and highly powerful machines are necessary to run the complete classification and grouping scheme in real time. Comparatively speaking, these two features have been efficiently implemented in the work described in this paper, by using a second order polynomial model for the road edges recursively updated under the least squares approach, and plausibly demonstrated running on a simple PC Pentium 120MHz in real time.

On the other hand, a neurally inspired approach was developed in the ALVINN system [31], providing the steering angle of the vehicle directly from the visual analysis of the incoming image processed by a neural network. A proper training phase was needed prior to achieving correct on-line performance in similar scenarios. Thus, during the training period the neural network supervisedly learns the knowledge from a set of images corresponding to some stretch of road. After the training algorithm converges, the neural system autonomously drives the vehicle in roads similar to those used in the training stage. ALVINN was successfully demonstrated on the streets of the Carnegie Mellon University Campus at moderate speeds. However, it suffers from the dependency problem of the taught training patterns, which is inherent to the neural network approach. Another problem is that ALVINN does not provide the lateral position of the vehicle within the lane, but instead, the steering angle is directly issued from the visual image. Both problems are properly treated in VIRTUOUS by following the paradigm of avoiding implicit learning, and so, removing the need for supervised learning, as well as by precisely calculating the lateral and orientation errors of the vehicle, as will be described later.

This leads us to conclude that VIRTUOUS is nowadays one of the most robust and accurate vision-based systems for autonomous navigation on unstructured roads. The validity and generality of VIRTUOUS is plausibly supported by real experiments in real conditions. In fact, autonomous navigation on a private circuit along hundreds of kilometres has been carried out during the last year using this paradigm, proving its robustness and appropriateness for this task.

Chapter 4

Navigation on Intersections

A complete navigation mission in a complex road network can be accomplished by properly arbitrating the execution of VIRTUOUS, for non-structured road tracking, and an efficient algorithm that correctly drives the vehicle along intersections and crossroads. This situation demands the design and implementation of an explicit vision-based procedure for intersection navigation, as the behaviour required for this task is largely different from that required for road tracking.

The problem of vision-based intersection recognition and navigation has been scarcely treated in the technical literature. Thus, in some cases the visual detection of intersections is simply aimed at providing the driver with audible signals for warning purposes [37]. The first system capable of navigating on intersections with limited success was developed at the Carnegie Mellon University under the Navlab project. The problem of recognition was restricted to Y-shaped intersections with a maximum angle of 50 degrees between their branches. As the authors recognise in their last publication [21], on one hand there is much work left to be done to robustly detect all roads and intersections, and on the other hand, the weak link of the system is its inability to navigate road junctions once they are found.

A more recent approach carried out at the Universität der Bundeswehr München (UBM) undertakes the challenge of intersection recognition and navigation on a network of unstructured roads [28]. By exploiting images from two monochrome cameras, intersections were detected and tracked utilising an active pan-tilt head (TACC) to direct the focus of attention. Building on these results, vision based intersection navigation has been integrated in an Expectation-based Multi-focal Saccadic Vision system (EMS-Vision), using a whole arrangement of four cameras. The EMS-Vision system has proved to some extent its ability to navigate on a non-complex network of unstructured roads. As part of their future work the authors of the EMS-Vision system are currently focusing on navigation on unpaved road networks with intersections with multiple branches.

In the present work intersection recognition is strictly carried out basing on geometric information provided by a DGPS receiver. A priori data is then necessary for this purpose, and thus a previous global map containing the X-Y coordinates of all intersections in the environment has been off-line constructed. On the other hand, intersection navigation is completely vision based, and accounts for any angular value between the intersection branches. A simple monocular colour vision system is proposed to realise intersection navigation (indeed the same system utilised for road tracking). This fact becomes a major issue, as cheap prototype navigation systems will be possible.

Two basic manoeuvres can be executed at an intersection: on one hand the vehicle can change its moving direction by turning left or right; on the other hand the vehicle can go ahead and cross the intersection by keeping its current direction. The problem of crossing an intersection is basically the same of tracking the lane, and thus the same algorithmic solution is provided for this kind of manoeuvre. On the contrary, turning right or left at an intersection is quite a different problem that needs to be addressed on a detailed chapter.

4.1 Turning manoeuvres at intersections

The navigation strategy proposed for turning manoeuvres at intersections resembles the human way of driving. Considering the limitation in the camera's field of view, the vehicle should start the turning manoeuvre (left or right, according to the plan) at low speed, until enough perspective of the new road is gained. From that point onwards lane tracking will resume control of the vehicle and its velocity will gradually increase. To gain better understanding of the complexity implicit in this kind of manoeuvres, consider the sequence of images depicted in figure 4.1, where the perspective of the road gradually improves as the vehicle turns to the left.

Figure 4.1 Sequence of images during a left turn.

Thus, perception efforts must be focused on the direction where the road is expected to appear upon turning completion. In other words, the right part of the image will be the focus of attention when the vehicle is performing a right turn at an intersection, for instance.

A remarkable circumstance to account for is the fact that road edges can no longer be modelled as parabolic polynomials during the turning manoeuvre, due to the limited perspective of the road particularly at the beginning of the turn. In this situation, there is no correspondence between the real road edges and a second order function, in particular in cases when even the edges are not visible within the camera's field of view. On the other hand, there exists a clear necessity to provide continuity in the road edges estimation, bearing in mind that the vehicle turning angle is issued from the estimation of the central trajectory of the road as will be described later, and so, there should be no abrupt discontinuity in the model estimation when traversing an intersection. This means that an estimation of the road edges is needed anyway.

To solve this contradictory and compromising situation (the estimation of road edges is needed but it can not rely on parabolic polynomials) we propose to de-couple the segmentation process from the road edges estimation. This leads to the use of a fixed road model that shall not be updated according to the image segmentation results during the vehicle turn. That invariable model will be the valid reference so as to obtain the vehicle turning angle, until enough perspective of the road is gained and road tracking can be resumed. Thus, segmentation and road edges estimation are made independent from each other during the turn at an intersection.

The fixed road model will be geometrically located in the image plane in the direction of the intended turn to perform at the intersection, according to the global plan. For instance, figure 4.2 shows the fixed models for left (a) and right (b) turns at an intersection. Reminding that the vehicle turning angle is obtained from the central trajectory of the road, models depicted in figure 4.2 will cause the vehicle to turn left or right, respectively. Besides, these models are located in the region of the image where the road is expected to appear upon turning completion, and thus, no discontinuity in the vehicle turning angle will occur. On the other hand, the road width estimation is also kept constant during the whole turn, assuming the last estimated value before starting the turn $W(t_0)$.

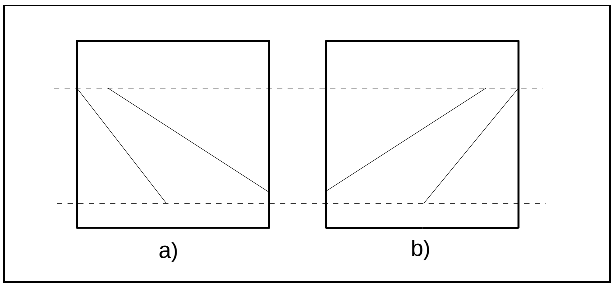


Figure 4.2 Fixed road models for left (a) and right (b) turns at intersections.

Continuity in the road edges and width estimation is preserved due to the least squares based estimator that brings the road model from its initial position, at the beginning of the turning manoeuvre (at $t = t_0$), to the corresponding fixed model in a soft and gradual manner.

As previously mentioned, the end of the turning manoeuvre is determined basing on visual information. For this purpose, the image segmentation is compared (or correlated) to several a priori road models, as will be described in detail later. The vehicle is supposed to appropriately perceive the new road when the previous correlation is high enough, i.e., when the new road resembles some of the a priori road models utilised in the comparison. From that point onwards, lane tracking resumes control of navigation. The process is coarsely described in the flow diagram depicted in figure 4.3.

4.1.1 Image processing at intersections

Basically, image processing at intersections is similar to image processing for road tracking, except for those processes concerning features adaptation as a function of the road model. Indeed, the road model can not be used for HSI updating during a turning manoeuvre at an intersection, as it remains fixed and thus provides no relevant information. Accordingly, segmentation is exclusively performed on the basis of HSI colour characteristics, and no threshold modification is carried

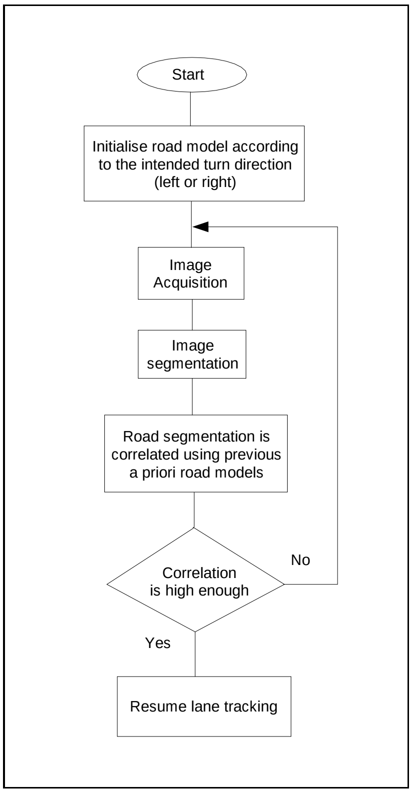


Figure 4.3 Basic algorithm for turning manoeuvres at intersections.

out basing on the distance between the pixel under consideration and the estimated road model. The rest of the image segmentation process is maintained invariable, yielding the sequence of segmented images depicted in figure 4.4 for a typical left turn.

Likewise, figure 4.5 illustrates the road edge estimation process in a sequence of images during a left turn at an intersection. As can be appreciated from observation of figure 4.5, the road model gradually shifts from quite a centred position (first image), through the fixed road model for left turns (images from 2 to 7), until road tracking is resumed when enough perspective of the new road is gained (image 8).

Finally, the HSI road colour pattern is updated basing on all pixels segmented as road at the current iteration (avoiding thus to rely on the road model, as it provides no reliable information during the turn). Experience demonstrates that the image segmentation and adaptation method presented in this section remains stable during intersection navigation, as far as enough road is perceived in the camera's field of view.

4.2 Determining the end of the turning manoeuvre

As previously described, the end of the turning manoeuvre is determined basing on visual information. In particular, the turning manoeuvre finishes when the new road is perceived with a sufficiently reliable perspective. This permits to navigate on any kind of intersection and to execute any possible turning angle.

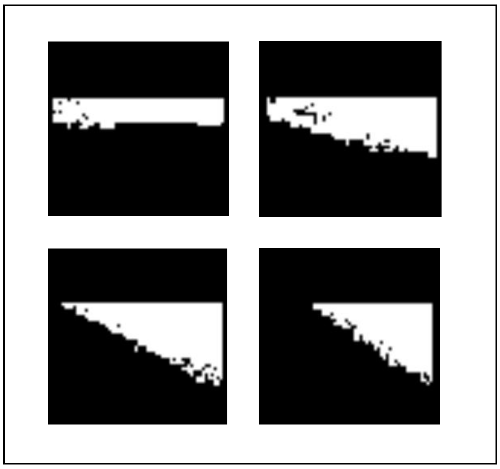


Figure 4.4 Segmented images during a left turn at an intersection.

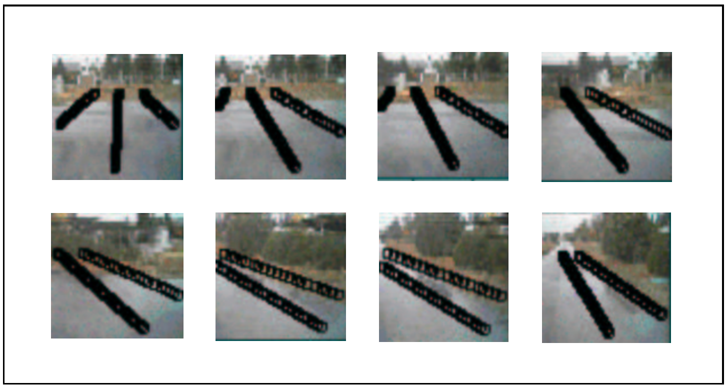


Figure 4.5 Road edge estimation during a left turn at an intersection.

On the other hand, the exclusive use of visual information leads to dangerous situations, as the visible portion of the road does not fit a second order polynomial (yielding quite noisy segmentations) due to the limited perspective of the road during turning manoeuvres. In a first intuitive approach, the end of the turning manoeuvre could be determined by computing the correlation between the incoming segmentation and some a priori road models. This model based template matching requires the definition of a priori road models, as depicted in figure 4.6 for the case of left turns.

Unfortunately, experience demonstrates that the use of this simple correlation measure does not suffice for reliable determination of the end of the turn. On one hand, segmentations similar to the a priori road model templates can occur even at the beginning of the turn, and thus a false detection should happen. On the other hand, segmentation noise drastically increases during the turn due to the absence of a parabolic model that contributes to enhancement purposes using spatial constraints. Relying on these facts, we propose to reinforce the vehicle localisation during the turn using Markov stochastic processes, in what will be referred to as Markov Localisation Process hereinafter. The basic idea is to enhance the vehicle localisation robustness while keeping on using visual information at the same time. The angular trajectory described by the vehicle during the turn is modelled by a random variable denoted by ξ, as depicted in figure 4.7.

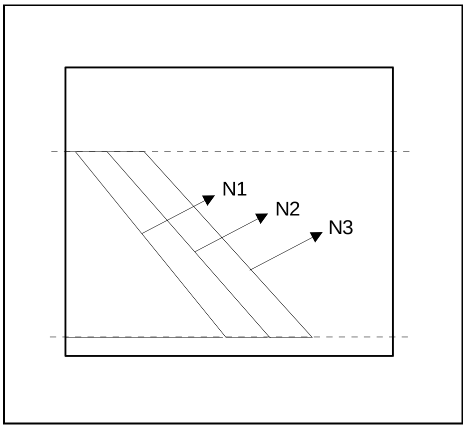


Figure 4.6 A priori road model templates for left turns.

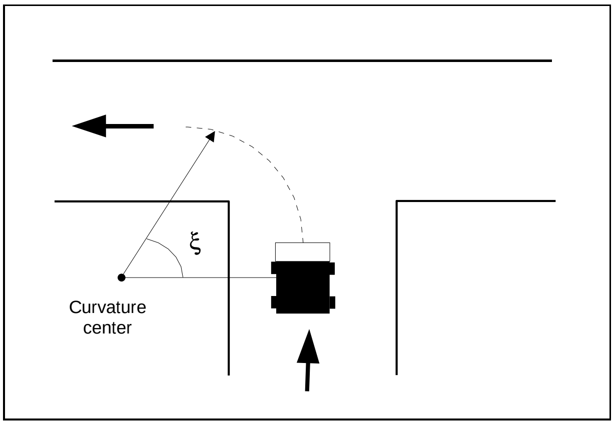


Figure 4.7 Modelling of the vehicle turning angle ξ at intersections.

A probability density function is calculated for all possible positions along the localisation space. Such a function is updated at each iteration time under the typical Markov assumptions, and in doing so it becomes a Markov stochastic process. The abcise ξ_{max} where the density function reaches its maximum indicates the most reliable vehicle angular position during the turn.

Let $Bel(\xi_t = \xi)$ denote the vehicle's belief of being at location ξ at time t, where ξ is a location in the localisation space. The possible values of ξ range from 0 degrees at the beginning of the turn, to 90 degrees or higher at the end of the turning manoeuvre. $Bel(\xi_0)$ reflects the initial state of knowledge. If the vehicle position is accurately known, $Bel(\xi_0)$ is centred on such location. If not so, $Bel(\xi_0)$ is uniformly distributed to reflect the global uncertainty in vehicle location. In this work, $Bel(\xi_0)$ is initially set to 0 degrees, as depicted in figure 4.8, taking advantage of the fact that the vehicle is starting the turn. The distribution $Bel(\xi)$ is updated whenever the vehicle moves or acquires a new image.

4.2.1 $Bel(\xi)$ updating upon vehicle movement

Vehicle motion is modelled by the conditional probability $p_a(\xi/\xi^{'})$. $p_a(\xi/\xi^{'})$ denotes the probability that motion action a, when executed at ξ', carries the vehicle to position ξ. $p_a(\xi/\xi^{'})$ is then

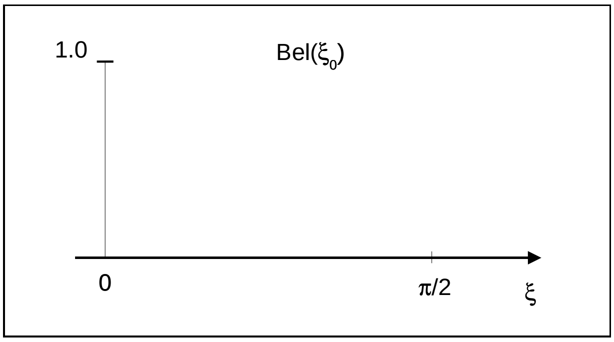


Figure 4.8 Initial belief distribution $Bel(\xi_0)$.

used to update the belief upon vehicle motion, where $\widehat{Bel}(\xi_t = \xi)$ denotes the resulting belief at time t, as indicated in equation 4.1.

$$\widehat{Bel}(\xi_t = \xi) = \sum_{\xi'} p_a(\xi/\xi') \cdot Bel(\xi_{t-1} = \xi') \tag{4.1}$$

Computation of $p_a(\xi/\xi')$ is carried out accounting for the vehicle kinematic and dynamic constraints. This implies the use of the vehicle kinematic model (approximated by the popular Ackermann model), and proprioceptive knowledge about vehicle current velocity v and steering angle ϕ. Let R denote the radius of curvature of the trajectory described by the vehicle during the turn, as depicted in figure 4.9. Basing on that figure, the differential angular arc $\Delta\xi$ described by the vehicle between two consecutive iterations, can straightforward be obtained considered that the vehicle linear velocity v is kept constant, according to equation 4.2.

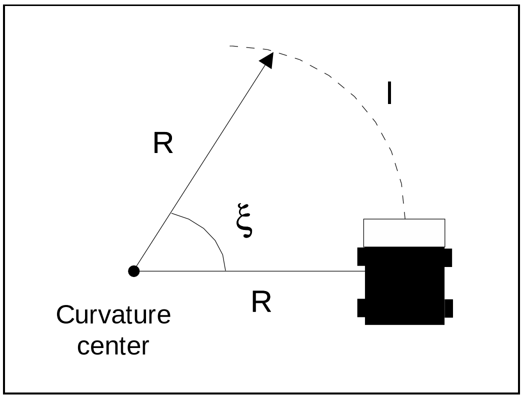


Figure 4.9 Vehicle radius of curvature.

$$\Delta\xi = \frac{\Delta l}{R} = \frac{v \cdot \Delta t}{R} \tag{4.2}$$

where Δt represents the time between two consecutive algorithm iterations, and v stands for the vehicle linear velocity. On the other hand, the radius of curvature R can be calculated using the vehicle kinematic model, yielding the expression in equation 4.3.

$$R = \frac{L}{\tan \phi} \tag{4.3}$$

where L denotes the wheelbase, and ϕ stands for the vehicle steering angle. Thus, $\Delta\xi$ can be explicitly written as a function of measurable magnitudes, as in equation 4.4.

$$\Delta\xi = \frac{v \cdot \Delta t \cdot \tan \phi}{L} \tag{4.4}$$

Although $\Delta\xi$ can not be regarded as an exact value, due to sliding, backlash, and measure noise not explicitly considered in the model, equation 4.4 is helpful in modelling $p_a(\xi/\xi^{'})$. The probability that the vehicle reaches position ξ at the current state, from ξ' at the previous one, can be understood as the probability that the vehicle covers an angular trajectory $\psi = \xi - \xi^{'}$ in a time interval Δt. Basing on the previously described model, $\Delta\xi$ is the most likely value for $p_a(\xi/\xi^{'})$. Likewise, the probability will gradually diminish as the difference between ξ and ξ' increases. On the other hand, the vehicle is physically constrained to move forward, and thus, the probability that its angular position decreases with time is null.

$$p_a(\xi/\xi^{'}) = 0 \qquad \forall \xi < \xi^{'} \tag{4.5}$$

According to the previous reasoning, we propose to model $p_a(\xi/\xi^{'})$ by means of an auxiliary random variable denoted by Γ. Probability $p_a(\xi/\xi^{'})$ can be substituted by probability $p_a(\Gamma = \psi)$, so that $p_a(\Gamma < 0) = 0$, while a gaussian function is considered for positive values of Γ, yielding a maximum at $\Gamma = \Delta\xi$. Graphically, the model for $p_a(\xi/\xi^{'})$ is depicted in figure 4.10.

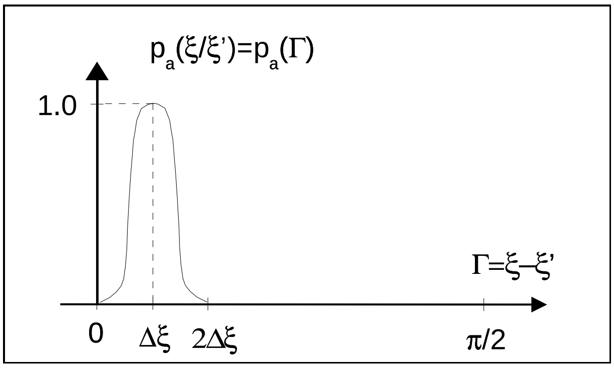


Figure 4.10 Model of $p_a(\xi/\xi^{'})$.

The standard deviation of the gaussian model for $p_a(\xi/\xi^{'})$ has been empirically set to $\sigma = \Delta\xi$, and so, the probability is almost zero for angular values near $\Gamma = 0$. Nevertheless, due to the fact that $p_a(\eta = \xi)$ is zero for negative values of Γ, the proposed model is not exactly a gaussian function. This effect is corrected by using a normalising factor $p(s)$ as described in the next section.

4.2.2 $Bel(\xi)$ updating upon image acquisition and processing

$Bel(\xi)$ distribution must be validated according to the visual information contained in the scene acquired by the vehicle vision system. Let s denote the vision system measure, representing the degree of similarity or correlation between the current image and some a priori road model expected to be perceived upon intersection completion. On the other hand, $p(s/\xi)$ stands for the probability of obtaining measure s at position ξ. The belief distribution $Bel(\xi)$ is updated upon image acquisition and processing according to equation 4.6.

$$Bel(\xi_t = \xi) = \frac{p(s/\xi) \cdot \widehat{Bel}(\xi_t = \xi)}{p(s)} \tag{4.6}$$

where $p(s)$ represents a normalising factor to ensure that $Bel(\xi)$ is truly a real probability density function (p.d.f). Visual measure s is obtained by computing the correlation between the incoming segmentation and several a priori road models, on a pixel by pixel basis. In particular, we've devised three a priori road models located on the area of the image where the road is most likely to appear after completing the turn at an intersection. Additionally, the width of the a priori road models is randomly chosen in a given interval around the road width $\hat{W}(t_0)$ estimated just before starting the intersection manoeuvring. This endows the system with the capacity to recognise and track roads with different widths. Figure 4.11 shows the shape of the a priori road models for left (a) and right (b) turns.

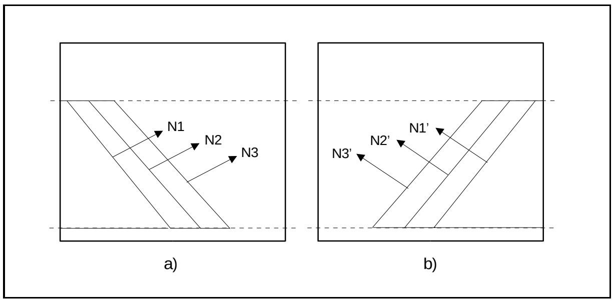


Figure 4.11 A priori road models for left (a) and right (b) turns at intersections.

For each a priori road model coefficients r_i and $\overline{r_i}$ are computed. Coefficient r_i measures the similarity between the road area in the segmented image and the road area in a priori model i. Likewise, $\overline{r_i}$ measures the correlation between the non road area in the segmented image and the non road area in a priori model i. These coefficients are calculated as shown in equation 4.7.

$$\begin{aligned} r_i &= \frac{N_{i-road}}{T_{i-road}} \\ \overline{r_i} &= \frac{N_{i-noroad}}{T_{i-noroad}} \end{aligned} \tag{4.7}$$

where T_{i-road} stands for the total number of road pixels in a priori model i, and $T_{i-noroad}$ represents the total number of non road pixels in such model. On the other hand, N_{i-road} is the number of road pixels in the segmented image that match the road pixels in a priori model i, while $N_{i-noroad}$ is the number of non road pixels in the segmented image matching the non road pixels in model i. Correlation index s is computed basing on the maximum value of r_i and $\overline{r_i}$, evaluated over the three a priori road models as shown in equation 4.8.

$$s = max_i \frac{r_i + \overline{r_i}}{2} \tag{4.8}$$

where s is in the range $0 \leq s \leq 1$. The modelling of conditional probability $p_a(s/\xi)$ is accomplished accounting for dynamic constraints in the vehicle steering system, and thus, the recovery turning manoeuvre should start a little before completely finishing the turn in order to anticipate the trajectory and avoid overshoot and oscillations. Accordingly, an experimental value $\xi_T = 70°$ degrees is established to indicate that the probability of starting the recovery turn manoeuvre greatly increases upon completing an angular trajectory $\xi_T \geq 70°$, whenever the correlation measure s validates the estimation.

An exact environment map should be used to precisely model $p_a(s/\xi)$. In some previous research concerning mobile robot localisation in indoor environments [46] probability $p_a(s/l)$ is precomputed (where l stands for the robot location), basing on a global map and a sensor model, and stored on a look-up table. The use of such look-up table permits online computation of $p_a(s/l)$ in a simple and fast process. This kind of technique is deployed for radar or laser based systems in reduced environments. Considering that none of the previous conditions are given in a vision-based system in large outdoor scenarios, the use of precomputation becomes quite a complex an inefficient task.

Instead, we propose a simple and intuitive modelling, successfully proved in practice, by which the probability of measuring a high value of s will be very low at the beginning of the turn, but increasingly higher as the angular trajectory of the vehicle gradually approaches ξ_T. Figure 4.12 depicts the exact model for $p_a(s/\xi)$.

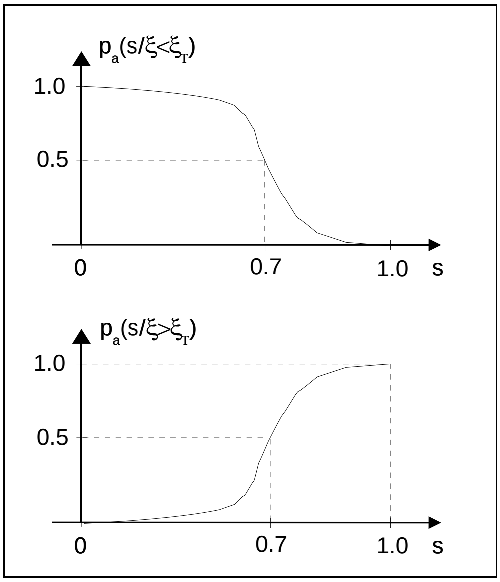


Figure 4.12 Modelling of $p_a(s/\xi)$.

Modelling of probability $p_a(s/\xi)$ has been split in two intervals. Thus, for any value smaller than ξ_T the probability of measuring a high correlation s is low, while the probability of obtaining a low correlation is gradually higher. On the contrary, for angles greater than ξ_T the vehicle is close to complete the turn and, accordingly, the probability of measuring a high correlation s increases, keeping a low value otherwise. Although $p_a(s/\xi)$ is not a real p.d.f (the integral of $p_a(s/\xi)$ along its definition domain does not sum 1.0) $Bel(\xi)$ distribution can indeed be assured to be a real p.d.f due to the normalising factor $p(s)$ in equation 4.6.

Considering that vehicle movement and image acquisition are simultaneous and continuously being carried out, $Bel(\xi)$ distribution will be updated at each iteration of the algorithm by consecutively applying equations 4.1 and 4.6. From the practical point of view, the definition domain of variable ξ must necessarily be discretised so as to make the problem computationally treatable. An angular resolution of $0.5°$ has been set for this purpose, providing a more than sufficient precision for approximate localisation. This implies that for a typical angular range in a turn of about 90°, the number of probabilities to compute amounts up to 180×180. However, most of the time probabilities different from zero are focused on a narrow interval. This observation permits to accomplish a Selective Computation that increases the algorithm execution speed, by only considering those angular values of chi for which probability $Bel(\Upsilon = \xi)$ is above some given threshold (1% of the maximum probability in this case).

The Markov localisation method described in this section provides a belief distribution $Bel(\xi)$ that tends to get a Gaussian shape. The average of such approximately Gaussian $Bel(\xi)$ represents the most likely vehicle location. In order to get a reliability measure of the estimated vehicle location, the belief distribution $Bel(\xi)$ is compared to a Gaussian function $N(\xi_{max}, \Delta\xi)$ (where ξ_{max} is the average of the gaussian function that best fits $Bel(\xi)$). The comparison is performed in the least squares sense as in equation 4.9.

$$\Xi = \frac{1}{N_\xi} \sum_\xi (Bel(\xi) - \frac{1}{\sqrt{2\pi}\Delta\xi} \exp^{\frac{-(\xi-\xi_{max})^2}{2\cdot\Delta\xi^2}})^2 \tag{4.9}$$

where Ξ represents the mean square error between distribution $Bel(\xi)$ and the gaussian function $N(\xi_{max}, \Delta\xi)$, and N_ξ is the number of points of the discrete definition domain of variable ξ. The comparison is graphically depicted in figure 4.13.

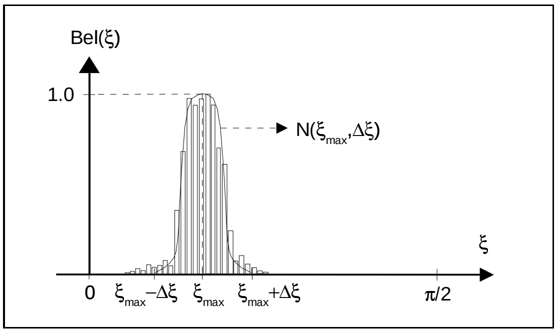
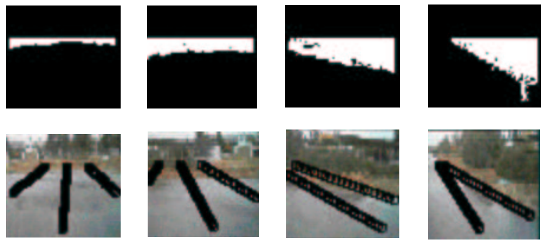


Figure 4.13 Comparison between distribution $Bel(\xi)$ and a gaussian function $N(\xi_{max}, \Delta\xi)$.

The vehicle should finish the turning manoeuvre and resume lane tracking when simultaneously the estimated angular position ξ_{max} is above ξ_T (i.e. close to the end of the turn), and Ξ is below some given threshold (experimentally set to 2.5) indicating a high confidence in the estimated vehicle location. To sum up, the Markov localisation method permits to statistically enhance visual measures, exhibiting an appropriate behaviour for vehicle localisation due to its ability to manage uncertainty and degrees of reliability.

4.3 Intersection navigation results

To illustrate the behaviour of the navigation system described in this section, figure 4.14 depicts a sequence of 4 real images during a left turn manoeuvre at an intersection. In that figure, we represent for each image the estimated road model overprinting the original incoming scene, the segmented image, and the values of s, ξ_{max} and Ξ.



s	0.51	0.62	0.69	0.85
ξ_{max} ($\circ$)	2	37	49	68
Ξ	2.47	2.39	2.33	2.28

Figure 4.14 Estimated road model and segmentation for a sequence of images in a left turn intersection.

As derived from observation of figure 4.14, the reliability about vehicle location remains high (i.e., a low value of Ξ) throughout the whole turning manoeuvre. Likewise, we can appreciate how the road model starts at the last location estimated during lane tracking before commencing the turn (first image of the sequence in figure 4.14), and gradually updates until reaching the fixed a priori road model devised for left turns (third image of the sequence in figure 4.14). When conditions for completing the turn are met ($\xi_{max} > \xi_T$ and $\Xi < 2.5$) lane tracking is resumed allowing to adapt the road model to the new road, as shown in the last image of the sequence in figure 4.14. To gain a better understanding of the global process, figure 4.15 depicts a complete example in which the vehicle is tracking a lane until it reaches an intersection. In that point, the vehicle performs a right turn manoeuvre and, finally, lane tracking is resumed. The estimated road model is shown overprinting the original picture for every image of the sequence in figure 4.15.

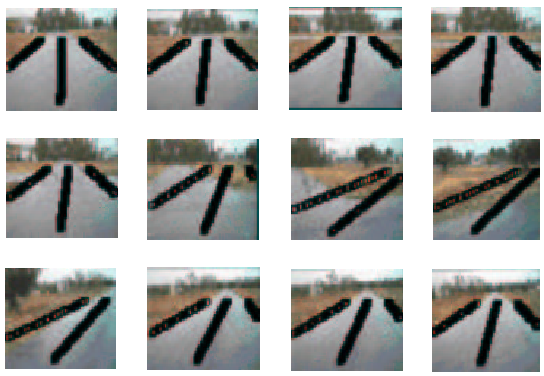


Figure 4.15 Estimated road model for a complete concatenation of actions: lane tracking-intersection navigation-lane tracking.

To sum up, the following concluding remarks must be pointed out. The navigation module proposed in this section provides continuity in the road model estimation, and ensures proper manoeuvring on intersections of arbitrary angular shape using one single colour camera. The localisation method accounts for the possibility of detecting roads with different width, after completing the turn at an intersection. As a major drawback, due to the limited perspective of the scene during turning manoeuvres at intersections, obstacle detection is not accurate enough making advisable the use of complementary sensors (radar or laser) to accomplish this task. Finally, navigation on typical urban roundabouts is considered as a future line of action.

Chapter 5

Vehicle detection

Obstacle detection is a basic safety skill every autonomous vehicle must be endowed with in order to achieve reliable navigation. Accurate road estimation as described in previous sections becomes an essential point in the obstacle detection process as it can help reduce the searching area to the limits of the estimated road. To gain some insight into the complexity of this vision based task just consider some real situations in an urban-like scenario such as absence of lane markers, parked cars on both sides of the street, zebra crossings, or other vehicles moving around. All these situations make it hard to reliably detect possible vehicles implying a danger for the ego-vehicle in the real world. According to the monocular colour vision system deployed in this work, we propose a specific vehicle detection module, leaving the development of a more general obstacle detection procedure for future work.

5.1 Searching Area

The execution time is reduced by limiting the vehicle detection area to the limits of the estimated road. This choice is strongly supported by the fact that vehicles out of the limits of the road don't make a real danger during navigation, and hence, the algorithm robustness is preserved. Nevertheless, the vehicle detection process relies considerably on the estimated road edges. This makes the algorithm not applicable in road sections where the estimation is not accurate enough (as in certain crossroads) but fully operative otherwise.

In order to robustly detect and track vehicles along the road, two consecutive processing stages are necessary, as depicted in figure 5.1. In the first step vehicles are localised basing on differential and symmetry properties, while in the second one the already detected vehicles are tracked using a real time estimator. A detailed description of both processes is given below.

5.2 Vehicle detection

Basically, the identification of other vehicles is performed according to vertical edge and symmetry characteristics, under the assumption that vehicles generally have artificial rectangular and symmetrical shapes that make their vertical edges easily recognisable from the rest of the environment.

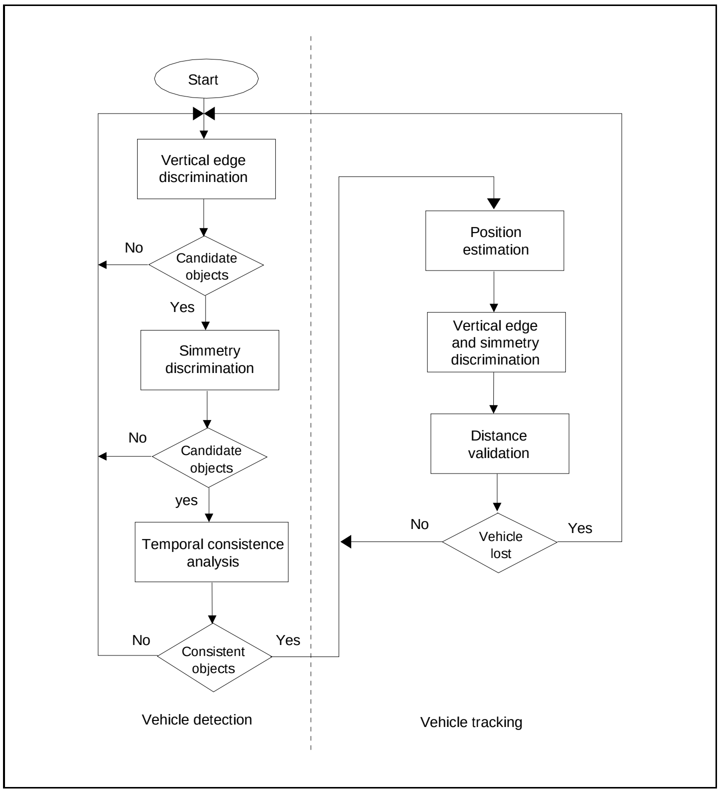


Figure 5.1 Algorithm for vehicle detection and tracking.

This is quite a realistic situation that can be reasonably assumed in practice.

5.2.1 Vertical edge and symmetry discriminating

A basic colour based vertical edge discriminating analysis is carried out on the area of interest. It permits to obtain candidate vertical edges representing the limits of the vehicles currently circulating on the road. Vertical edges are then considered in pairs so as to account only for couples that represent possible vehicle contours, disregarding those combinations that lead to unrealistic vehicle shapes.

In a second step, the discriminating process is refined according to symmetry constraints. Accordingly, regions located between two vertical edge pairs are studied in order to compute a vertical symmetry index. Only those regions complying with some given symmetry condition will be validated as candidate vehicles. To illustrate the process, figure 5.2 depicts an example of candidate vehicle detection based on vertical edge and symmetry features, as previously described.

5.2.2 Temporal coherence

Using spatial features as the only criterion for detecting vehicles yields to sporadic incorrect vehicle detection in real situations, due to environmental noise. Hence, a temporal validation filter becomes necessary to remove non-coherent objects from the scene. This means that an object val-

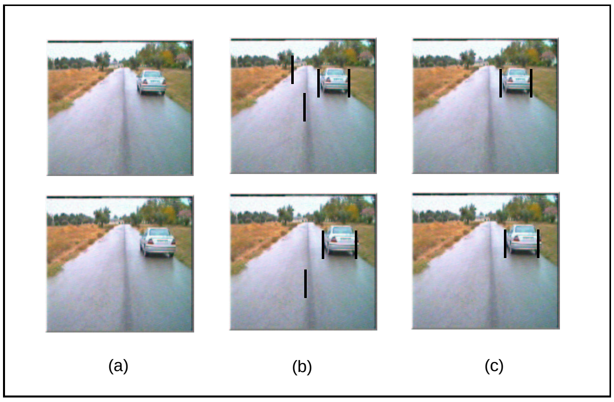


Figure 5.2 Examples of vehicle detection based on vertical edge and symmetry analysis. a) Original images. b) Vertical edges. c) Vertical edges candidates after symmetry analysis.

idated under the spatial features criterion must appear several consecutive iterations in the image in order to be considered as a real vehicle. Otherwise it is discarded and removed. A value has been used in practice to ensure that a vehicle appears in the image in a coherent time sequence.

During the time-spatial validation stage a major problem is to identify the appearance of the same vehicle in two consecutive frames. Vehicle identification is then carried out by making use of its position in correlative frames. In other words, the position differences permit to describe the evolution of the object in the image plane. At time instant the position of each validated object under the spatial criterion is annotated in a dynamic list and a time count is started in order to keep track of temporal coherence of all candidate vehicles. At time the process is repeated using the same spatial validation criterion. The time count is increased only for those objects whose distance to some of the previous candidate vehicles is less than . Otherwise the time count is reset. A candidate object is validated as a real vehicle when its time count reaches 5 consecutive iterations. Considering that the complete execution time of the vision algorithm is 100 ms, and the fact that the vehicle is operating in an urban scenario at velocities under 50 km/h, this empirical value has proven successfully to effectively detect real vehicles in the scene.

5.3 Vehicle Tracking

For each detected vehicle in the previous stage (over a maximum of two, one for each lane) an estimation process is started in order to keep track of the vehicle position. In a first step the new position of the vehicle is measured and validated according to geometric criteria. In the second step the position is filtered using a least squares estimator as described below.

5.3.1 Position validation

Vertical edge and symmetry features are calculated for every object in the incoming image, using the same validation criterion previously described. After that, data association for position validation is carried out using the (x, y) location of the validated objects. Basically it must be determined whether some of the objects in the current frame corresponds to some of the vehicles under tracking. Thus, the closest validated object to each vehicle must be found. For this purpose, the minimum distance to the objects is calculated around the estimated vehicle trajectory. This permits to account for the vehicle's movement direction, and hence, the searching process is enhanced by rejecting associations between objects and vehicles located in different lanes.

After having associated one candidate object to each detected vehicle a minimum distance criterion is used to validate the association. Thus, the Euclidean distance between the vehicle detected in the previous iteration and its current associated object must be under (the same validation distance used in the temporal coherence analysis) so as to improve noise rejection. In practice, there are many real situations like sudden brightness or shadows on the pavement, cluttered background noise, ... etc, that can derive in sporadic vertical edges yielding wrong candidate vehicles. Under these circumstances we could find that none of the candidate vehicles passes the position validation process. In such a case, the previous estimated position is kept for each vehicle under tracking. After (5 iterations) without obtaining any validated position we consider that the vehicle has disappeared from the scene (we've lost track of it). If this happens, vehicle tracking is stopped and the vehicle detection stage is started again.

5.3.2 Position estimation

A recursive least squares estimator with exponential decay is the key element to filter vehicle position measures after the association and validation processes. This filtering technique permits to keep an estimate of the vehicle position in iterations where no valid measure has been obtained. Obviously, tracking of each vehicle is performed using independent estimators.

Let be the estimated vehicle position in the previous iteration, where x and y represent the vertical and horizontal vehicle coordinates in the image plane, respectively. Likewise, let be the vehicle coordinates measured in the current iteration after validation. The current estimated vehicle position, given by state vector $x(t)$, is calculated in three consecutive steps as indicated in equations 5.1 to 5.4.

a)Update prediction.

$$\hat{z}(t) = x(t-1) \tag{5.1}$$

b)Update state covariance estimate.

$$P(t) = \frac{1}{\lambda'} \cdot [P(t-1) - K(t) \cdot P^T(t-1)] \tag{5.2}$$

c)Update state estimate.

$$x(t) = x(t-1) + K(t) \cdot [z(t) - \hat{z}(t)] \tag{5.3}$$

being

$$K(t) = P(t-1) \cdot [\lambda' I + P(t-1)]^{-1} \tag{5.4}$$

where $P(t)$ represents an estimate of the state covariance and λ' is a scalar parameter (ranging from 0 to 1) that allows to keep a trade off between estimation robustness and dynamic response. λ' has been empirically set to 0.75. The real vehicle coordinates in the 3D space are calculated from the estimated position in the image plane basing on the flat terrain assumption, as described in [9].

To illustrate the vehicle detection algorithm, figure 5.3 shows a real situation in which a vehicle is detected along the opposite lane in a sequence of several images. Likewise, figure 5.4 depicts another example, where a vehicle is detected along the same lane. The position of the detected vehicle is indicated by a black squared box in both examples.

Figure 5.3 Vehicle detection along the opposite lane.

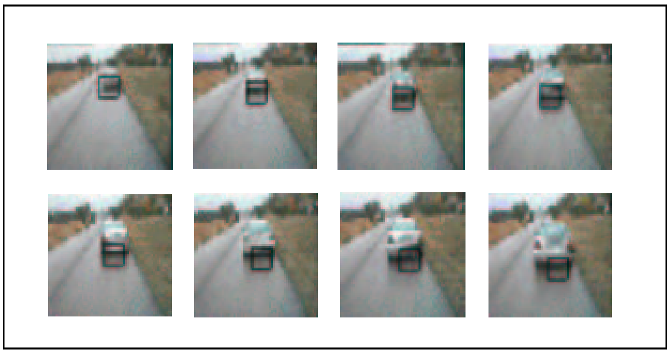


Figure 5.4 Vehicle detection along the same lane.

5.4 Adaptive Navigation

After detecting the presence of a vehicle, we proceed to determine the lane where it is located. Accordingly, proper actions must be taken in order to ensure safe navigation. Thus, if the detected vehicle is located along the opposite lane, the ego-vehicle should basically keep on driving along its own lane and so, no basic modification of the velocity or steering direction must be carried out. On the other hand, if a vehicle is detected along the same lane the ego-vehicle velocity is modified in an adaptive cruise control manner to keep some safety distance (in this work, we have established a safety distance $D_s = 10m$).

Let v_1 be the ego-vehicle velocity at the current iteration (measured from a tachometer system) and v_2 the estimated velocity of the detected vehicle. Velocity v_2 is estimated assuming, for simplicity, that it is kept constant during the time interval Δt between two consecutive iterations of the algorithm. Let d_1 and d_0 be the distances measured to the preceding vehicle in the previous and current iterations of the algorithm, respectively. An estimation of v_2 is carried out as indicated in equation 5.5.

$$v_2 = v_1 - \frac{d_1 - d_0}{\Delta t} \tag{5.5}$$

To keep a safety distance , the reference velocity must be updated as described in equation 5.6.

$$v_1 = v_2 - \frac{D_s - d_o}{\Delta t} \tag{5.6}$$

Further navigation actions are undertaken, involving steering commands, if the preceding vehicle's velocity decreases to a low value (under 20 km/h) or in case it stops. Under these circumstances a change lane manoeuvring is started whenever enough space to pass is detected from the visual analysis of the scene.

Chapter 6

Vehicle Control

Vehicle control has required the efforts and attention of many research groups around the world. Most of them explicitly focus on automatic steering control [7] leaving aside longitudinal control issues, as the latter is indeed quite a more simple problem. Nonetheless, both velocity (longitudinal) and steering (lateral) control have been implemented in this work in order to provide completely autonomous operation.

6.1 Longitudinal Control

The longitudinal control module enables the vehicle to keep the reference velocity established in the global velocity profile, as computed by the global planner at the beginning of the autonomous mission. That reference speed will be kept unless the previously mentioned safety distance is violated due to the presence of other vehicles in the same lane.

Figure 6.1 shows the complete velocity control scheme, where a simple but robust fuzzy controller [**?**] has been designed for this purpose, providing excellent dynamic response upon step changes in the reference velocity. In fact, figure 6.2 depicts the vehicle response after applying increasing and decreasing reference steps in the velocity profile.

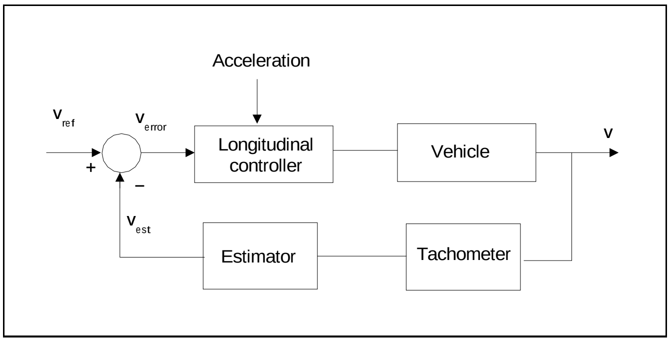


Figure 6.1 Velocity control scheme.

As observed in figure 6.2, the longitudinal controller successfully demonstrates to suffice for reliable velocity control in real circumstances.

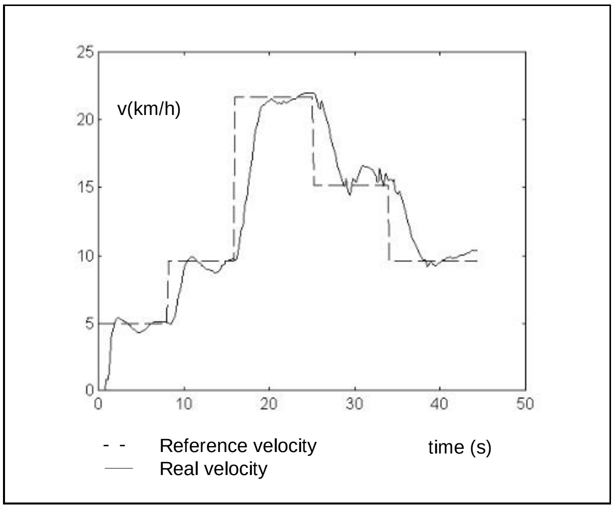


Figure 6.2 Velocity response upon step changes in the reference profile.

6.2 Lateral Control

The main goal of the lateral control module is to ensure proper tracking of the road by correctly keeping the vehicle in the centre of the lane with the appropriate orientation (parallel to the road trajectory). This constraint can be translated into the minimisation of the lateral and orientation errors $((d_e)\ (\theta_e))$ as illustrated in figure 6.3.

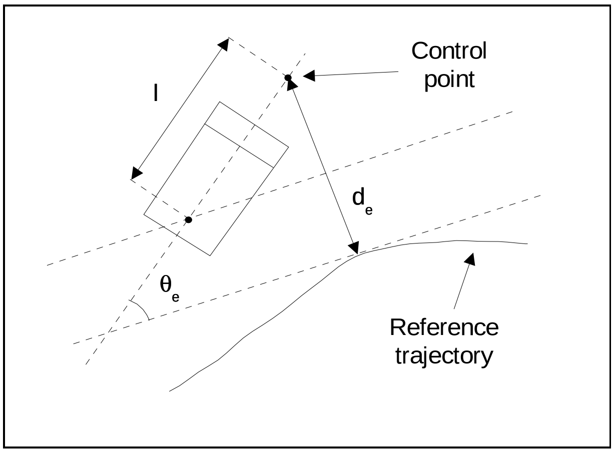


Figure 6.3 Lateral and orientation errors with respect to a reference trajectory.

To solve this controllability problem and design a stable lateral controller, a model describing the dynamic behaviour of d_e and θ_e is needed, as well as an adequate error measure system.

6.2.1 Kinematic model

The kinematic model of the vehicle is the starting point to model the dynamics of the lateral and orientation errors. The vehicle model is approximated by the popular Ackerman (or bicycle) model [8] as depicted in figure 6.4, assuming that the two front wheels turn slightly differentially and thus, the instantaneous rotation centre can be purely computed by kinematic means.

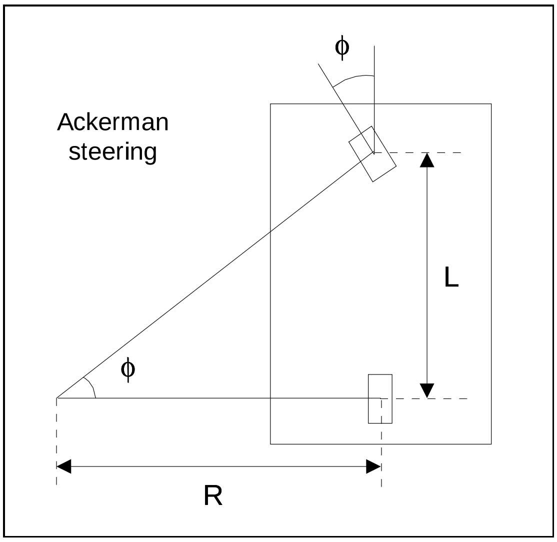


Figure 6.4 Approximate kinematic model of the vehicle (Ackerman steering).

Let $\kappa(t)$ denote the instantaneous curvature of the trajectory described by the vehicle.

$$\kappa(t) = \frac{1}{R(t)} = \frac{\tan\phi(t)}{L} = \frac{d\theta(t)}{ds} \tag{6.1}$$

where R is the radius of curvature, L is the wheelbase, ϕ is the steering angle, and θ stands for the vehicle orientation in a global frame of coordinates. The dynamics of theta is computed in equation 6.2 as a function of vehicle velocity v.

$$\dot{\theta} = \frac{d\theta}{dt} = \frac{d\theta}{ds} \cdot \frac{ds}{dt} = \kappa(t) \cdot v(t) = \frac{\tan\phi(t)}{L} \cdot v(t) \tag{6.2}$$

Let ϕ and v be the variables of the vehicle actuation space. On the other hand, the vehicle configuration space is composed of the global position and orientation variables, described by (x, y, θ), under the flat terrain assumption. Mapping from the actuation space to the configuration space can be solved by using the popular Fresnel equations, which are also the so-called dead reckoning equations typically used in inertial navigation. Equation 6.3 shows the dynamics of (x, y, θ).

$$\begin{array}{c} \dot{x} = \frac{dx}{dt} = v(t)\cos\theta(t) \\ \dot{y} = \frac{dy}{dt} = v(t)\sin\theta(t) \\ \dot{\theta} = \frac{d\theta}{dt} = v(t)\frac{\tan\phi(t)}{L} \end{array} \tag{6.3}$$

where $v(t)$ represents the velocity of the midpoint of the vehicle rear axle, denoted as control point. Nonetheless, global information about the position and orientation of the vehicle (x, y, θ) is of little use for a vision based system that can only compute local information in the incoming scene. Thus, the development of a dynamic model for the lateral and orientation errors becomes necessary.

As observed in figure 6.3, the lateral error d_e is defined as the distance between the vehicle control point and the closest point along the vehicle desired trajectory, described by coordinates (x_d, y_d). This implies that d_e is perpendicular to the tangent to the reference trajectory at (x_d, y_d). The scope of the tangent at (x_d, y_d) is denoted by θ_d, and represents the desired orientation at that point. Basing on this, d_e and θ_e are proved to suffice to precisely characterise the location error between the vehicle and some given reference trajectory, as described in equations 6.4 and 6.5.

$$d_e = -(x - x_d) \cdot \sin\theta_d + (y - y_d) \cdot \cos\theta_d \tag{6.4}$$

$$\theta_e = \theta - \theta_d \tag{6.5}$$

Computing the derivative of d_e with respect to time yields to equation 6.6, while the time derivative of θ_e is shown in equation 6.7. Thus, the complete non-linear dynamic model for d_e and θ_e is formulated in equation 6.8.

$$\dot{d}_e = -\dot{x}\sin\theta_d + \dot{y}\cos\theta_d = -V\cos\theta\sin\theta_d + V\sin\theta\cos\theta_d = V\sin(\theta - \theta_d) = V\sin\theta_e \tag{6.6}$$

$$\dot{\theta}_e = \frac{d(\theta - \theta_d)}{dt} = \dot{\theta} - \dot{\theta}_d = \dot{\theta} \tag{6.7}$$

$$\begin{aligned} \dot{d}_e &= V\sin\theta_e \\ \dot{\theta}_e &= \tfrac{V}{L}\tan\phi \end{aligned} \tag{6.8}$$

6.2.2 Non-linear control law

The control objective is to ensure that the vehicle will correctly track the road that visually perceives. For this purpose, both the lateral error d_e and the orientation error θ_e must be minimised. On the other hand, vehicle velocity v will be assumed to be constant according to the global velocity profile, for simplicity.

The design of the control law is based on general results in the so-called chained systems theory. An excellent example on this topic can be found in [11]. Nevertheless, these results are extraordinarily extended and generalised in this paper so as to provide a stable non-linear control law for visual road tracking.

From the control point of view, the use of the popular tangent linearisation approach is avoided as it is only valid locally around the configuration chosen to perform the linearisation, and thus, the initial conditions may be far away from the reference trajectory. On the contrary, some state and control variables changes are posed in order to convert the non linear system described in equation 6.8 into a linear one, without any approximation (exact linearisation approach). Nevertheless, due to the impossibility of exactly linearising systems describing mobile robots dynamics, we can convert these non linear systems in almost linear ones, termed as chained form. The use of the chained form permits to design a control law using to a high extent linear systems theory.

In particular, the non-linear dynamic model for d_e and θ_e (equation 6.8) can be transformed into chained form using the state diffeomorphism and change of control variables, as in equation 6.9.

$$
\begin{aligned}
Y &= \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \Theta(X) = \begin{bmatrix} d_e \\ \tan\theta_e \end{bmatrix} \\
W &= \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \Upsilon(U) = \begin{bmatrix} v\cos\theta_e \\ \frac{v\tan\phi}{L\cos^2\theta_e} \end{bmatrix}
\end{aligned}
\tag{6.9}
$$

These transformations are invertible whenever the vehicle speed v is different from zero. From equation 6.9 the vehicle dynamic model can be rewritten as in equation 6.10, considering y_1 and y_2 as the new state variables.

$$
\begin{aligned}
\dot{y}_1 = \dot{d}_e = v\sin\theta_e = w_1 y_2 \\
\dot{y}_2 = \frac{d(\tan\theta_e)}{dt} = \frac{1}{\cos^2\theta_e}\cdot\dot{\theta}_e = \frac{v\tan\phi}{L\cos^2\theta_e} = w_2
\end{aligned}
\tag{6.10}
$$

In order to get a velocity independent control law, the time derivative is replaced by a derivation with respect to ς, the abscissa along the tangent to the reference trajectory as graphically depicted in figure 6.5.

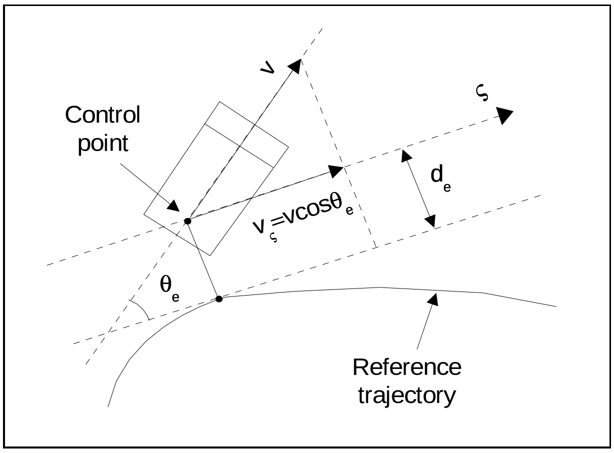


Figure 6.5 Graphical description of ς.

Analytically, ς is computed as the integral of velocity v_ς, measured along axis ς.

$$
\varsigma = \int v_\varsigma dt = \int v\cos\theta_e dt \quad \Rightarrow \quad \dot{\varsigma} = \frac{d\varsigma}{dt} = v\cos\theta_e = w_1 \tag{6.11}
$$

The time derivative of the state variables y_1 and y_2 is expressed as a function of ς in equation 6.12.

$$
\begin{aligned}
\dot{y}_1 = \frac{dy_1}{dt} = \frac{dy_1}{d\varsigma}\cdot\frac{d\varsigma}{dt} = y_1'\cdot\dot{\varsigma} \\
\dot{y}_2 = \frac{dy_2}{dt} = \frac{dy_2}{d\varsigma}\cdot\frac{d\varsigma}{dt} = y_2'\cdot\dot{\varsigma}
\end{aligned}
\tag{6.12}
$$

where $y_1^{'}$ and $y_2^{'}$ stand for the derivative of y_1 and y_2 with respect to ς, respectively. Solving for $y_1^{'}$ and $y_2^{'}$ yields to equation 6.13.

$$\begin{aligned} y_1^{'} = \frac{\dot{y}_1}{\dot{\varsigma}} = \frac{v\sin\theta_e}{v\cos\theta_e} = \tan\theta_e = y_2 \\ y_2^{'} = \frac{\dot{y}_2}{\dot{\varsigma}} = \frac{v\tan\phi}{L\cos^2\theta_e v\cos\theta_e} = \frac{\tan\phi}{L\cos^3\theta_e} = \frac{w_2}{w_1} = w_3 \end{aligned} \tag{6.13}$$

As observed in the previous equation, the transformed system is linear and thus, state variables y_1 and y_2 can be regulated to zero (so as to yield $d_e = d_{e,ref} = 0$ and $\theta_e = \theta_{e,ref} = 0$) by using the control low proposed in equation 6.14.

$$w_3 = -K_d y_2 - K_p y_1 \qquad (K_d, K_p) \in \Re^{+2} \tag{6.14}$$

Using equations 6.13 and 6.14 and solving for variable y_1 yields to equation 6.15, where the dynamic behaviour of y_1 with respect to ς is demonstrated to be linear.

$$y_1^{''} + K_d y_1^{'} + K_p y_1 = 0 \tag{6.15}$$

This implies that variables $y_1 = d_e$ and $y_2 = \tan\theta_e$ tend to zero as variable ς grows. The previous statement is analytically expressed in equation 6.16.

$$\lim_{\varsigma\to\infty} d_e = \lim_{\varsigma\to\infty} \theta_e = 0 \tag{6.16}$$

Accordingly, variable ς must always grow so as to ensure that both d_e and θ_e tend to zero. This condition is met whenever $v > 0$ and $-\pi/2 < \theta_e < \pi/2$. In other words, the vehicle must continuously move forward and the absolute value of its orientation error should be below $\pi/2$ in order to guarantee proper trajectory tracking. Thus, the non-linear control law is finally derived from equation 6.13 and 6.14.

$$\phi = \arctan[-L\cos^3\theta_e \cdot (K_d\tan\theta_e + K_p d_e)] \tag{6.17}$$

The control law is then modified by a sigmoidal function as shown in equation 6.18, to account for physical limitations in the vehicle wheels turning angle and prevent from actuator saturation. On the other hand, the use of sigmoidal functions preserves the system stability as demonstrated in [**?**].

$$\phi = \arctan\left[-KL\cos^3\theta_e \cdot \frac{1-\exp^{-K(K_d\tan\theta_e+K_p d_e)}}{1+\exp^{-K(K_d\tan\theta_e+K_p d_e)}}\right] \tag{6.18}$$

The control law is saturated to ϕ_{max} by properly tuning parameter K. Thus, the maximum value of equation 6.18 is $\phi_{max} = \pm\arctan(-KL)$. Therefore, K is chosen to ensure that $\phi_{max} = \pm\frac{\pi}{6} rad$ (physical limitation of the vehicle), given the wheelbase $L = 2.69m$, yielding a practical value $K = 0.2146$.

$$K = \frac{\tan\frac{\pi}{6}}{L} \tag{6.19}$$

From observation of equation 6.15, the dynamic response of variable y_1 can be considered to be a second order linear one. In practice, it is not indeed linear due to the sigmoidal function used to saturate the control law, although it can be reasonably approximated as such. Thus, an analogy between constants K_d , K_p, and the parameters of a second order linear system ξ (damping coefficient) and ω_n (natural frequency) can be established, yielding equation 6.20.

$$\begin{aligned} \omega_n &= \sqrt{K_p} \\ \xi &= \frac{K_d}{2\sqrt{K_p}} \end{aligned} \tag{6.20}$$

Likewise, system overshoot M_p and settling distance d_s (given that the system dynamics is described as a function of space variable ς, not time) can be obtained from equation 6.21.

$$\begin{aligned} M_p &= \exp^{\frac{-\xi\pi}{\sqrt{1-\xi^2}}} \\ d_{s|2\%} &= \frac{4}{\xi\omega_n} \end{aligned} \tag{6.21}$$

The design of constants K_d and K_p is undertaken considering that the system overshoot must not exceed 10% of the step input, and that the settling distance should be below some given threshold. Thus, for a typical settling time $t_s = 20s$, and given the vehicle velocity v, the proper settling distance can be computed as in equation 6.22.

$$d_s = t_s \cdot v = 20v \tag{6.22}$$

The value of K_d is derived from equations 6.20 and 6.21 yielding the velocity dependant expression in equation 6.23.

$$K_d = \frac{8}{d_s} = \frac{0.4}{v} \tag{6.23}$$

Likewise, dumping coefficient ξ is derived from equations 6.20 and 6.21, as shown in equation 6.24.

$$\xi = \sqrt{\frac{1}{[\frac{\pi}{\ln 0.1}]^2 + 1}} = \frac{K_d}{2\sqrt{K_p}} = \frac{4}{d_s\sqrt{K_p}} \tag{6.24}$$

Finally, K_p is deduced from the previous equation, yielding equation 6.25.

$$K_p = \left[\frac{6.766}{d_s}\right]^2 = \left[\frac{0.3383}{v}\right]^2 \tag{6.25}$$

The dependence of K_p and K_d on vehicle velocity v permits to ensure proper dynamic response. In particular, vehicle turning angle will be soft at high speeds, therefore avoiding possible oscillations due to physical constraints in the steering dynamics.

6.2.3 Extension of the control law for high speeds

The non-linear control law designed in the previous section provides stable trajectory tracking at moderate speed (up to 10-20 km/h) in urban environments. However, experience demonstrates that tracking errors and vehicle oscillation increase as velocity rises. It becomes necessary then to develop an extension of the non-linear control law for high speeds. On the other hand, a stable controller at high speed will permit not only to drive the vehicle at typical velocities in urban environments (up to 50 km/h) but also to deploy the control system in automated highway vehicles.

The first step is to modify the vehicle control point as depicted in figure 6.6, in order to anticipate the road curvature at a given distance L_h denoted by Look-ahead distance. The new lateral and orientation errors are then computed as illustrated in the same figure 6.6, yielding the results in equation 6.26.

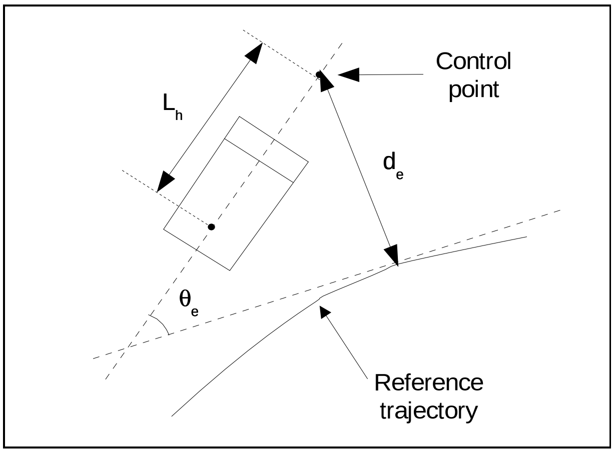


Figure 6.6 Lateral and orientation errors at the Look-ahead distance.

$$\begin{array}{c} d_e = -(x + L_h \cos\theta - x_d)\sin\theta_d + (y + L_h \sin\theta_d - y_d)\cos\theta_d \\ \theta_e = \theta - \theta_d \end{array} \tag{6.26}$$

The choice of L_h is carried out basing on the current vehicle velocity v, as described in [6], yielding the parameters shown in equation 6.27.

$$L_h(v) = \begin{cases} L_{min} \quad if \quad v < v_{min} \\ vt_1 \quad if \quad v_{min} \leq v \leq v_{max} \\ L_{max} \quad if \quad v > v_{max} \end{cases} \tag{6.27}$$

where $t_1 = 1.5s$ is the look-ahead time, $v_{min} = 25km/h$, $v_{max} = 75km/h$, $L_{min} = 10.41m$, and $L_{max} = 31.25m$. Considering the same scheme followed in the previous section, the new non-linear dynamic model for d_e and θ_e is shown in equation 6.28.

$$\begin{array}{c} \dot{d}_e = v\sin\theta_e + \frac{vL_h}{L}\cos\theta_e \tan\phi \\ \dot{\theta}_e = \frac{v\tan\phi}{L} \end{array} \tag{6.28}$$

This model can be transformed into chained form using the state diffeomorphism and change of control variables, as in equation 6.29.

$$
\begin{array}{c}
Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \Theta(X) = \begin{bmatrix} d_e \\ \tan\theta_e \end{bmatrix} \\
W = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \Upsilon(U) = \begin{bmatrix} v\cos\theta_e + \frac{vL_h \cos^2\theta_e \tan\phi}{L\sin\theta_e} \\ \frac{v\tan\phi}{L\cos^2\theta_e} \end{bmatrix}
\end{array}
\quad (6.29)
$$

These transformations are invertible whenever the vehicle speed v is different from zero. From equation 6.28 the vehicle dynamic model can be rewritten as in equation 6.30, considering y_1 and y_2 as the new state variables.

$$
\begin{array}{c}
\dot{y}_1 = \dot{d}_e = v\sin\theta_e + \frac{vL_h}{L}\cos\theta_e\tan\phi = w_1 y_2 \\
\dot{y}_2 = \frac{d(\tan\theta_e)}{dt} = \frac{1}{\cos^2\theta_e}\cdot\dot{\theta}_e = \frac{v\tan\phi}{L\cos^2\theta_e} = w_2
\end{array}
\quad (6.30)
$$

In order to get a velocity independent control law, the time derivative is replaced by a derivation with respect to ς, a variable related to the abscissa along the tangent to the reference trajectory. Analytically, ς is computed according to the following expression.

$$
\varsigma = \int (v\cos\theta_e + \frac{vL_h\cos^2\theta_e\tan\phi}{L\sin\theta_e} \quad (6.31)
$$

The time derivative of the state variables y_1 and y_2 is expressed as a function of ς in equation 6.32.

$$
\begin{array}{c}
\dot{y}_1 = \frac{dy_1}{dt} = \frac{dy_1}{d\varsigma}\cdot\frac{d\varsigma}{dt} = y_1^{'}\cdot\dot{\varsigma} \\
\dot{y}_2 = \frac{dy_2}{dt} = \frac{dy_2}{d\varsigma}\cdot\frac{d\varsigma}{dt} = y_2^{'}\cdot\dot{\varsigma}
\end{array}
\quad (6.32)
$$

where y_1' and y_2' stand for the derivative of y_1 and y_2 with respect to ς, respectively. Solving for y_1' and y_2' yields to equation 6.33.

$$
\begin{array}{c}
y_1^{'} = \tan\theta_e = y_2 \\
y_2^{'} = \frac{\tan\phi}{L\cos^3\theta_e + L_h\frac{\cos^4\theta_e\tan\phi}{\sin\theta_e}} = w_3
\end{array}
\quad (6.33)
$$

As in the previous section, the transformed system is linear and thus, state variables y_1 and y_2 can be regulated to zero (so as to yield $d_e = d_{e,ref} = 0$ and $\theta_e = \theta_{e,ref} = 0$) by using the new control low proposed in equation 6.34.

$$
w_3 = -K_d y_2 - K_p y_1 \qquad (K_d, K_p) \in \Re^{+2} \quad (6.34)
$$

Using equations 6.33 and 6.34 and solving for variable y_1 yields to equation 6.35, where the dynamic behaviour of y_1 with respect to ς is demonstrated to be linear.

$$
y_1^{''} + K_d y_1^{'} + K_p y_1 = 0 \quad (6.35)
$$

Once again, this implies that variables $y_1 (= d_e)$ and $y_2 (= \tan\theta_e)$ tend to zero as variable ς grows. The previous statement is analytically expressed in equation 6.36.

$$\lim_{\varsigma \to \infty} d_e = \lim_{\varsigma \to \infty} \theta_e = 0 \tag{6.36}$$

Accordingly, variable ς must always grow so as to ensure that both d_e and θ_e tend to zero. This condition is met whenever $v > 0$ and $-\pi/2 < \theta_e < \pi/2$. In other words, the vehicle must continuously move forward and the absolute value of its orientation error should be below $\pi/2$ in order to guarantee proper trajectory tracking. Basically, stability conditions remain the same as in the previous section. Thus, the new non-linear control law for high speeds is finally derived from equations 6.33 and 6.34.

$$\phi = \arctan\left[\frac{-L\sin\theta_e\cos^3\theta_e(K_d\tan\theta_e + K_p d_e)}{\sin\theta_e + L_h\cos^4\theta_e(K_d\tan\theta_e + K_p d_e)}\right] \tag{6.37}$$

The control law is then modified by a sigmoidal function to account for physical limitations in the vehicle wheels turning angle and prevent from actuator saturation. From this point onwards, tuning of K, K_d, and K_p follows the same scheme derived in equations 6.19, 6.23, and 6.25, respectively.

6.2.4 Lateral and orientation error measure

Lateral and orientation errors must be measured from the visual information contained in the 2D scene. For this purpose, the central trajectory of the road is assumed to be the vehicle reference trajectory for road tracking, and so, its 3D geometry is reconstructed under the flat terrain assumption.

We take a total of $N_c = 5$ measures belonging to the central trajectory of the road in the 2D image plane, and compute their corresponding coordinates in the 3D world (considering $Z = 0$, under the previously mentioned flat terrain assumption) using the popular pinhole camera model and the calibration parameters. Basing on the previous measures, a second order polynomial is then calculated to describe the central trajectory of the road in the 3D scene under the least squares approach, yielding equation 6.38.

$$X = AY^2 + BY + C \tag{6.38}$$

where the origin of the reference frame for axes X, Y, and Z is located at the midpoint of the vehicle rear axle, as depicted in figure 6.7.

For simplicity, the lateral error d_e is approximated as the difference between the reference trajectory and the Y-axis at the look-ahead distance, as in equation 6.39.

$$d_e = X(Y)_{|Y=L_h} = AL_h^2 + BL_h + C \tag{6.39}$$

Similarly, orientation error θ_e is measured as the relative difference between the orientation θ of the longitudinal vehicle axle and the orientation θ_d of the tangent to the reference trajectory, measured at the look-ahead distance L_h. Considering that the reference frame is fixed with regard to the vehicle rear axle, vehicle orientation θ is assumed to be zero, yielding the approximate orientation error shown in equation 6.40.

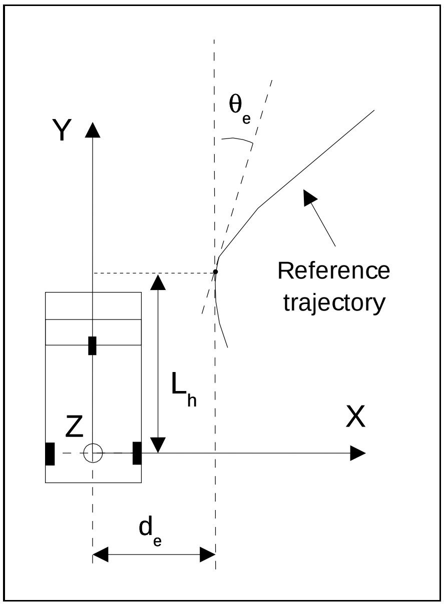


Figure 6.7 Reference frame for lateral and orientation error measure.

$$\theta_e = \theta - \theta_d = -\arctan \frac{dX}{dY}_{|Y=L_h} = -\arctan(2a_1 L_h + a_2) \tag{6.40}$$

6.2.5 Results

The complete close-loop scheme for lateral control is depicted in figure 6.8. As can be observed, the control objective is to achieve the reference error vector $d_{e,ref} = 0$, $\theta_{e,ref} = 0$. This objective implies proper tracking of the road curvature perceived by the vision system.

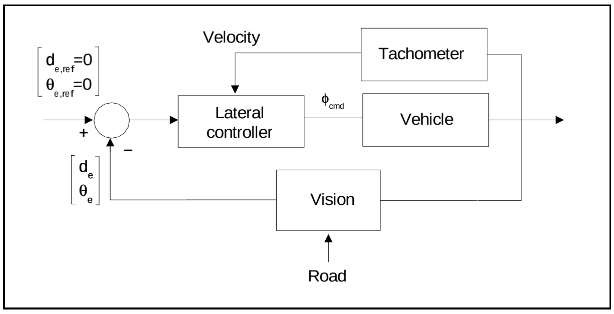


Figure 6.8 Close-loop lateral control scheme.

Various practical trials were conducted so as to test the validity of the control law for different initial conditions in real circumstances. During the tests, the reference vehicle velocity is assumed to keep constant by the velocity controller. Constants K_d and K_p were calculated as a function of v using equations 6.23 and 6.25.

Figures 6.9, 6.10, and 6.11 show the transient response of the vehicle lateral and orientation error for reference velocities of 10 km/h, 20 km/h, and 50 km/h respectively. In all cases, the vehicle starts the run at an initial lateral error of about 1m, and an initial orientation error in the range $\pm 5^\circ$. As can be clearly appreciated, the steady state response of the system is satisfactory for the three experiments. Thus, the lateral error is bound to $\pm 5cm$ at low speeds and $\pm 25cm$ at v=50 km/h, while the absolute orientation error in steady state remains below 1° in all cases.

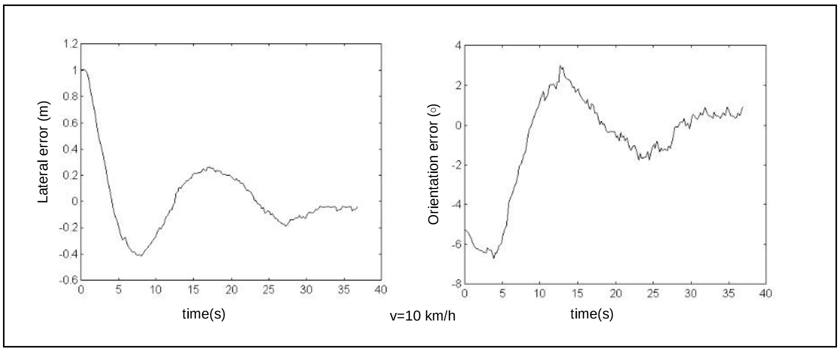


Figure 6.9 Transient response of the lateral and orientation error for v=10 km/h.

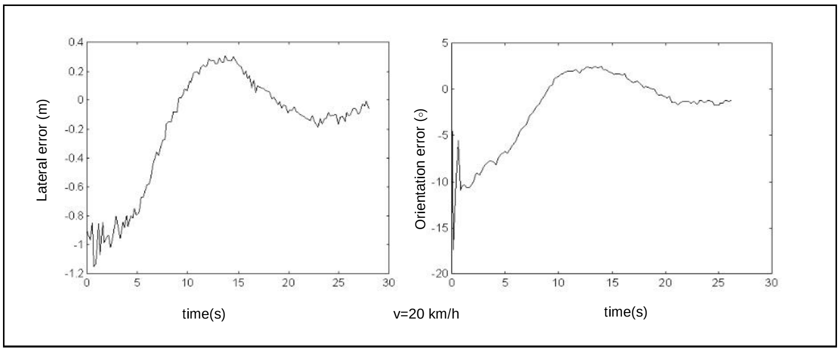


Figure 6.10 Transient response of the lateral and orientation error for v=20 km/h.

In a final test, the results achieved in the second test for v=20 km/h are compared to human driving at the same speed along the same trajectory. For this purpose a human driver steered the vehicle, leaving the control of the accelerator to the velocity controller in order to keep a reference speed of 20 km/h. The comparison is graphically depicted in figure 6.12.

On one hand, one can observe how the human driver takes less time than the automatic controller to achieve lateral and orientation errors close to zero. On the other hand, the steady state errors are similar in both cases. Surprisingly, human driving results in sporadic separations from the reference trajectory up to 40-50 cm, without incurring in dangerous behaviour, while the automatic controller keeps the vehicle under lower lateral error values once stabilised. Far from being an isolated fact, this circumstance was repeatedly observed in several practical experiments. As conclusion, the lateral control law developed in this work can reasonably be considered to be valid to drive a vehicle as precisely as a human can.

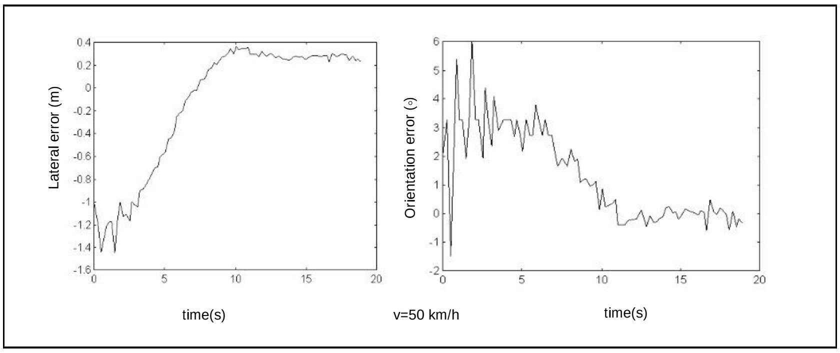


Figure 6.11 Transient response of the lateral and orientation error for v=50 km/h.

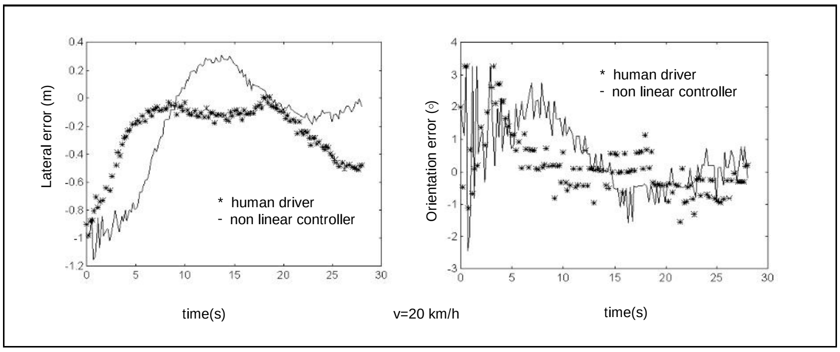


Figure 6.12 Comparison between automatic guidance and human driving at v=20 km/h.

Chapter 7

Implementation and Results

The complete navigation system described in the previous sections has been implemented on the so-called Babieca[1] prototype vehicle , an electric commercial Citroën Berlingo as depicted in figure 7.1, that has been modified to allow for automatic velocity and steering control at a maximum speed of 90 km/h.

Figure 7.1 Babieca prototype vehicle.

Babieca is equipped with a colour camera, a DGPS receiver, a Pentium PC, and a set of electronic devices to provide actuation over the accelerator and steering wheel, as well as to encode the vehicle velocity and steering angle. The colour camera provides standard PAL video signal at 25 Hz that is processed by a Meteor frame grabber installed on a 120 MHz Pentium running the Real Time Linux operating system. On the other hand, the DGPS receiver is a Z-12 Real Time model by Ashtech that implements the RTCM SC 104 V2.2 standard at 5 Hz. After implementing the complete navigation system under Real Time Linux using a pre-emptive scheduler [2], the lane tracking vision based task gets executed at 10 frames/s, while intersection navigation can be run at 4-5 frames/s.

Practical experiments were conducted on a private circuit located at the *Instituto de Automática Industrial* in Arganda del Rey (Madrid). The circuit is composed of several stop stations, streets, intersections, and roundabout points, trying to emulate an urban quarter. Although the graphical description of the circuit was provided in section 2 we depict it again in figure 7.2 for completeness reasons, indicating the real length of each street.

[1]Babieca was the horse of El Cid, a mythical Spanish hero that fought against the Arabs in the Middle Ages. In the battle of Valencia, Babieca on its own led the dead body of El Cid against the enemies contributing to a decisive victory

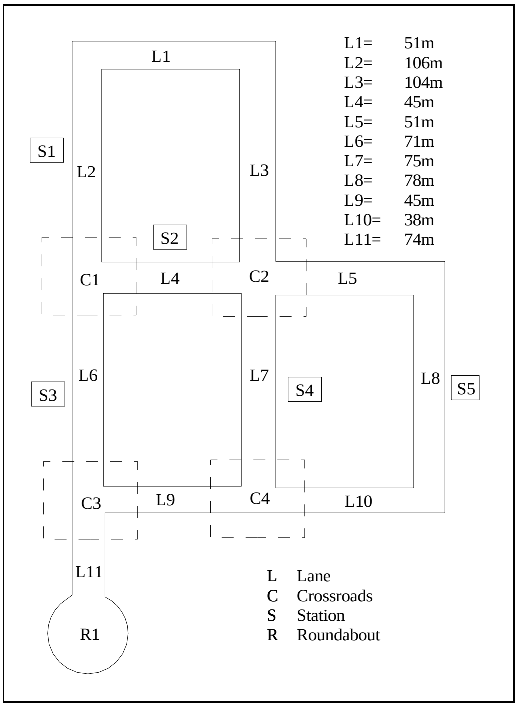


Figure 7.2 Test circuit map.

During the last year, Babieca ran over hundreds of kilometres in dozens of successful autonomous missions carried out along the test circuit. To illustrate the global behaviour of the complete navigation system implemented on Babieca some general results are shown next. Thus, in a first test the vehicle was told to autonomously navigate from station 1 to station 2. Figure 7.3 shows the 2D real trajectory followed by Babieca using UTM coordinates. Likewise, the vehicle real velocity and steering angle during the mission are depicted in figure 7.4, clearly showing the strong turns performed at intersections.

Similarly, figures 7.5 and 7.6 show the global trajectory accomplished by the vehicle to go from station 5 to station 1, and from station 4 to station 3, respectively.

All previous tests were conducted in the absence of other vehicles in the circuit. To put the vehicle detection system under test we introduce a second vehicle, driven by a human in this case, circulating at a lower speed along the same circuit in such a manner that it purposely interferes the pre-programmed vehicle path during an autonomous mission from station 1 to station 2. Figure 7.7 shows the new global trajectory followed by the autonomous vehicle during the mission, while figure 7.8 depicts the real velocity and steering angle during the mission in presence of an obstacle as described.

As can be observed from figure 7.7, the presence of a second vehicle ahead of the automatic vehicle does not affect the execution of the global trajectory according to the general plan. In fact, 2D trajectories shown in figures 7.3 (mission from station 1 to station 2 without obstacles) and 7.7 (mission from station 1 to station 2 in presence of an obstacle) are almost identical. On the other hand, vehicle velocity decreases suddenly with respect to the velocity profile during the first stretch of the mission, as depicted in figure 7.8a, due to the presence of a second vehicle circulating at low speed (15 km/h) along the autonomous vehicle path. Once the second vehicle pulls aside, the

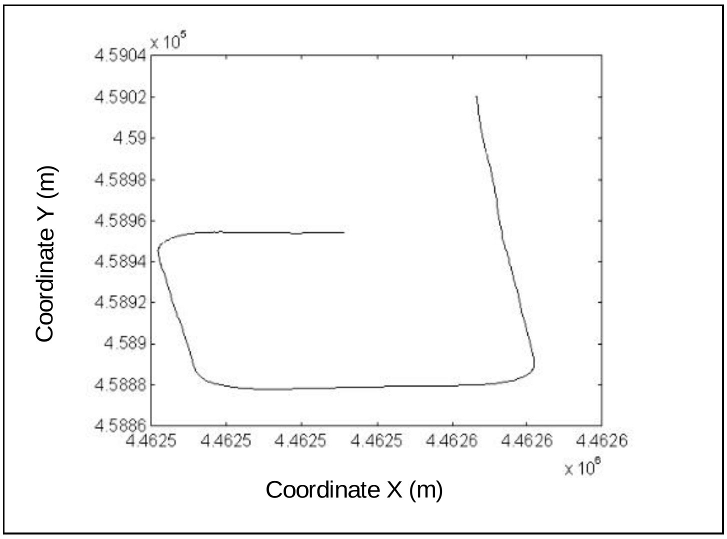


Figure 7.3 Real trajectory followed by the vehicle in an autonomous mission between stations 1 and 2.

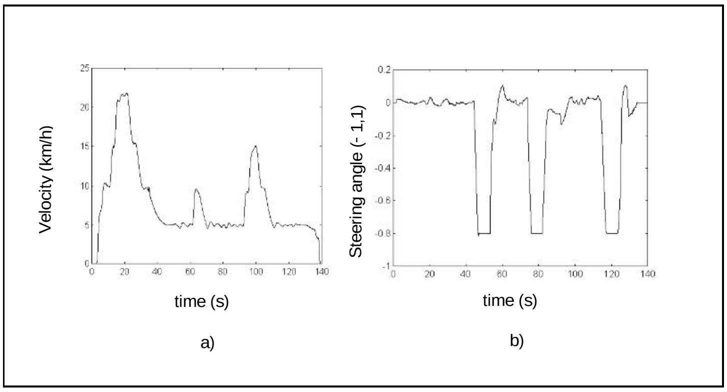


Figure 7.4 Autonomous mission from station 1 to station 2. a) Vehicle velocity. b) Vehicle steering angle.

autonomous vehicle increases its speed to reach again the velocity profile calculated by the global planner at the beginning of the mission, and normal operation is resumed. A complete set of video files demonstrating the operational performance of the system in real tests can be retrieved from ftp://www.depeca.uah.es/pub/vision.

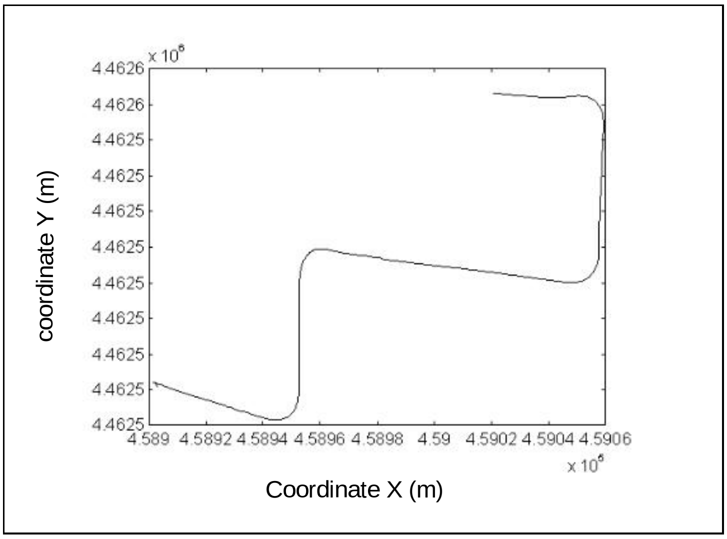


Figure 7.5 Real trajectory followed by the vehicle in an autonomous mission between stations 5 and 1.

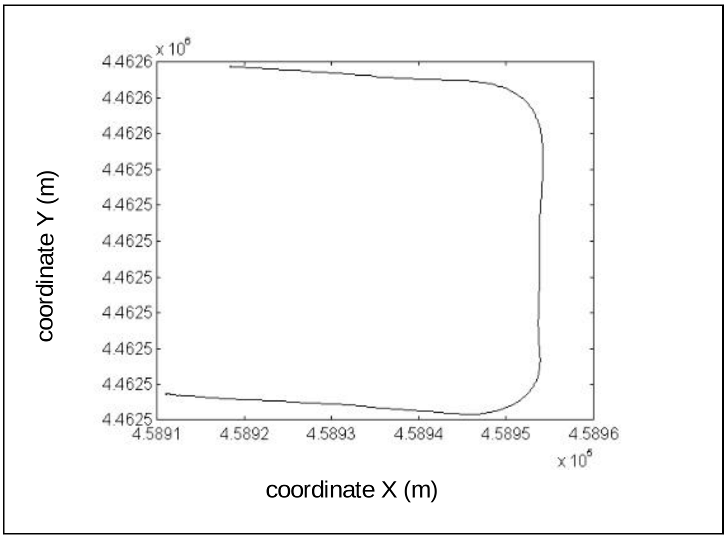


Figure 7.6 Real trajectory followed by the vehicle in an autonomous mission between stations 4 and 3.

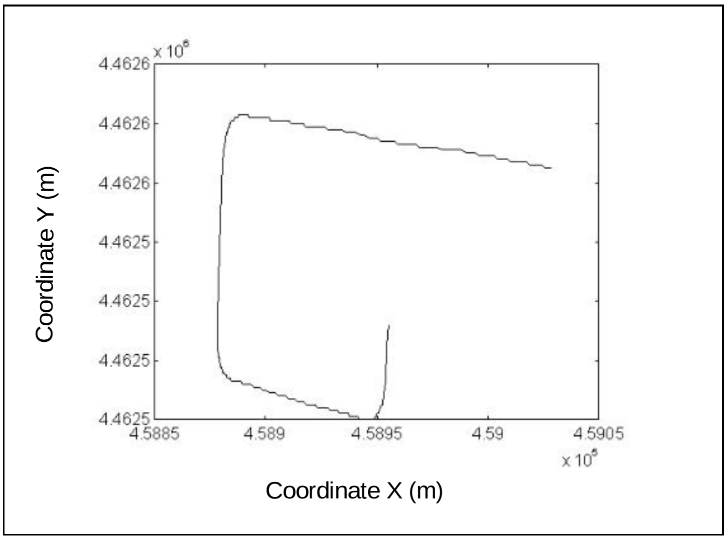


Figure 7.7 Real trajectory followed by the vehicle in an autonomous mission between stations 1 and 2, in presence of an obstacle.

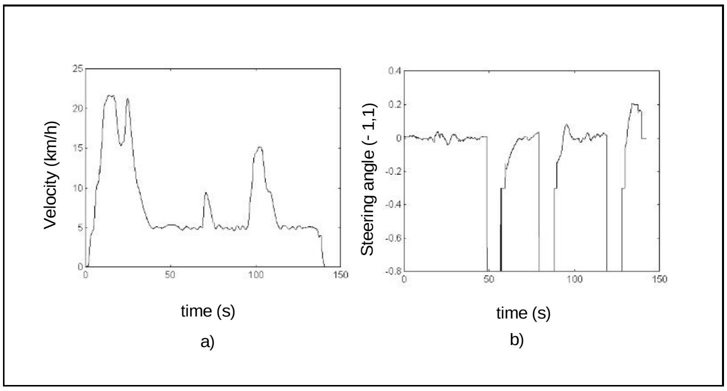


Figure 7.8 Autonomous mission from station 1 to station 2 in presence of an obstacle. a) Vehicle velocity. b) Vehicle steering angle.

Chapter 8

Conclusions and future work

To conclude, the next key points should be remarked. First of all, the vision and DGPS based global navigation system described in this work is currently implemented on a real commercial vehicle slightly modified so as to allow for autonomous operation. The complete system has been successfully tested on a private circuit, as a first step towards its long-term deployment on urban scenarios.

According to mission specifications and the a priori map, the global navigation system implements two complementary vision based behaviours for road tracking and navigation on intersections, respectively, in a coordinated manner. Thus, a task manager properly synchronises the execution of the adequate vision based task depending on whether the vehicle is running on a road, or it is traversing an intersection, making use of the DGPS for this purpose. The fact that no extremely high precision is needed in the DGPS signal, together with the use of a single colour camera results in a low cost final system, suitable for midterm commercial development.

On the other hand, proper tracking of non-structured roads is a major contribution of this work, as it is robust, not requiring previous learning, and allows for real time operation. Likewise, vision based intersection navigation is also a remarkable point due to its complexity and necessity to achieve continuity in navigating on a network of roads. This kind of manoeuvre is sparsely treated in the technical literature.

To our knowledge, the global navigation scheme implemented in this work is one of the first vision based systems in the world capable of performing autonomous missions in a network of non-structured roads, together with the work developed in [28].

Nonetheless, a lot of work still remains to be done until a completely robust and reliable autonomous system can be fully deployed in real conditions, as can obviously be thought. Thus, in the next step the vehicle detection module will be improved by combining information provided by other sensors, laser or radar based. Another key point is to remove the DGPS dependence by implementing a vision based task for intersection detection, permitting to use a conventional GPS receiver. Finally, another vision based specialised task will be developed in the future, intended to navigate not only in intersections, but in roundabouts, as the presence of these elements is wide spread in urban environments.

Acknowledgements

This work has been financed by the *Departamento de Electrónica* of the UAH, and the generous support of the *Instituto de Automática Industrial* of the CICYT.

Bibliography

[1] R. Alami, S. Fleury, M. Herrb, F. Ingrand and F. Robert. “Multi-Robot Cooperation in the MARTHA Project”. IEEE Robotics & Automation Magazine. March 1998.

[2] M. Barabanov. *A Linux-based Real-Time Operating System*. Master of Science Thesis in Computer Science. New Mexico Institute of Mining and Technology. New Mexico. June 1997.

[3] M. Bertozzi and A. Broggi. “GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection”. IEEE Transactions on Image Processing 7 (1). 1998. 62-81.

[4] M. Bertozzi, A. Broggi and A. Fascioli. “Vision-based intelligent vehicles: State of the art and perspectives”. Robotics and Automation Systems 32. 2000. 1-16.

[5] A. Broggi and S. Bertè. “Vision-based road detection in automotive systems: A real-time expectation-driven approach”. Journal of Artificial Intelligence Research 3. 1995.

[6] A. Broggi, M. Bertozzi, A. Fascioli and G. Conte. *Automatic Vehicle Guidance: the Experience fo the ARGO Autonomous Vehicle*. World Scientific. 1999.

[7] A. Broggi, S. Nichele, M. Bertozzi, and A. Fascioli. “Stereo Vision-based Vehicle Detection”. In Proceedings IEEE Intelligent Vehicles Symposium. Detroit, USA. October 2000.

[8] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi. “Shape-based Pedestrian Detection”. 2001.

[9] S. Cameron and P. Proberdt. *Advanced Guided Vehicles. Aspects of the Oxford AGV Project*. Ed. by World Scientific Publishing Co. Pte. Ltd. 1994.

[10] R. Chapuis, R. Aufrère and F. Chausse. “Recovering the 3D shape of a road by vision”. Proceedings of the International Conference on Image Processing Applications. Manchester, UK. July 1999. 686-690.

[11] P. Charbonnier, P. Nicolle, Y. Guillard and J. Charrier. “Road boundaries detection using color saturation”. Proceedings of the Ninth European Signal Processing Conference ’98. September 1998.

[12] L. Cordesses, P. Martinet, B. Thuilot and M. Berducat. “GPS-based control of a land vehicle”.

[13] J. D. Crisman and C. E. Thorpe. “UNSCARF, a color vision system for the detection of unstructured roads”. Proceedings of the IEEE international Conference on Robotics and Automation. Sacramento. 1991.

[14] T. De Pedro, R. García, C. González, J. E. Naranjo, J. Reviejo and M. A. Sotelo. “Vehicle Automatic Driving System Based on GNSS”. Proceedings of the International Conference on Intelligent Vehicles. Seville, 2001.

[15] S. Denasi, C. Lanzone, P. Martinese, G. Pettiti, G. Quaglia and L. Viglione. “Real-time system for road following and obstacle detection”. Proceedings of the SPIE on Machine Vision Applications, Architectures, and Systems Integration III. 1994.

[16] E. D. Dickmanns and A. Zapp. “A Curvature-based Scheme for Improving Road Vehicle Guidance by Computer Vision”. Proc. SPIE on Mobile Robots, vol 727. 1986.

[17] E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrant, M. Mauer, F. Thomanek, and J. Shielhlen. “The seeing passenger car ’VaMoRs-P”’. In Proc. of Int. Symp. on Intelligent Vehicles, Paris. 1994.

[18] E. D. Dickmanns. “Vehicles Capable of Dynamic Vision”. 15^{th} International Joint Conference on Artificial Intelligence (IJCA-97). Nagoya, Japan, August 23-29, 1997.

[19] U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Paitzold and C. Wöhler. “Autonomous driving goes downtown”. Proceedings of the IEEE Intelligent Vehicles Symposium. Stuttgart, Germany. 1998.

[20] A. Giachetti, M. Campani and V. Torre. “The use of Optical Flow for Road Navigation”. IEEE Transactions on Robotics and Automation. Vol 14. No 1, February 1998.

[21] R. C. Gonzales and R. E. Wood. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.

[22] M. H. Hebert, C. Thorpe and A. Stentz. *Intelligent Unmanned Ground Vehicles*. Autonomous Navigation Research at Carnegie Mellon. 1997.

[23] N. Ikonomakis, K. N. Plataniotis and Venetsanopoulos. “Color Image Segmentation for Multimedia Applications”. Journal of Intelligent and Robotic Systems. 2000.

[24] W. D. Jones. “Keeping cars from crashing”. IEEE Spectrum. September 2001.

[25] K. J. Kim, S. Y. Oh, S. W. Kim, H. Jeong, C. N. Lee, B. S. Kim and C. S. Kim. “An autonomous land vehicle PRV II: Progresses and performance enhancement”. Proceedings of the IEEE Intelligent Vehicles Symposium. Detroit. 1995.

[26] K. Kluge and C. Thorpe. “Intersection Detection in the YARF Road Following System”. Ingelligent Autonomous Systems 3. Pittsburgh, PA, USA. 1993.

[27] W. Li, X. Jiang and Y. Wang. “Road recognition for vision navigation of an autonomous vehicle by fuzzy reasoning”. Fuzzy Sets and Systems 93. Elsevier. 1998.

[28] M. Lützeler and E. D. Dickmanns. “Road recognition with MarVEye”. Proceedings of the IEEE Intelligent Vehicles Symposium. Stuttgart, Germany. 1998.

[29] M. Lützeler and E. D. Dickmanns. “EMS-Vision: Recognition of Intersections on Unmarked Road Networks”. International Symposium on Intelligent Vehicles (IV’2000). Dearborn, (MI). October 2000.

[30] H. P. Moraveck. “Towards automatic visual obstacle avoidance”. Proceedings of the 5th International Joint Conference on Artificial Intelligence. August 1977.

[31] V. Morellas, T. Morris, L. Alexander and M. Donath. “Between the lines: a driver-assistive technology for staying in the lane”. Intelligent Transportation Systems. GPS World. May 1998.

[32] D. A. Pomerleau. *Neural Network Perception for Mobile Robot Guidance*. Kluwer Academic Publishers: Boston. 1993.

[33] D. A. Pomerleau and T. M. Jockem. “Rapidly adapting machine vision for automated vehicle steering”. IEEE Expert 11 (2). 1996.

[34] K. A. Redmill. “A simple vision system for lane keeping”. Proceedings of the IEEE International Conference on Intelligent Transportation Systems. Boston. MA. 1997.

[35] J. Reviejo, T. de Pedro, J. E. Naranjo, R. García, C. González, and S. Alcalde. “An Open Frame to Test Techniques and Equipment for Autonomous Vehicles”. IFAC Symposium on Manufacturing, Modeling, Management and Control. University of Patras, Rio, Greece, July 2000, pp 350-354.

[36] R. Risack, P. Klausmann, W. Krüger and W. Enkelmann. “Robust lane recongnition embedded in a real-time driver assistance system”. Proceedings of the IEEE Intelligent Vehicles Symposium. Stuttgart, Germany. 1998.

[37] F. J. Rodríguez, M. Mazo and M. A. Sotelo. “Automation of an Industrial Fork Lift Truck, Guided by Artificial Vision in Open Environments”. Autonomous Robots, 5, 215-231. 1998. Kluwer Academic Publishers.

[38] S. Rossle, V. Kruger and Gengenbach. “Real-Time Vision-Based Intersection Detection for a Driver’s Warning Assistant”. Proceedings Intelligent Vehicle’93. Tokyo, Japan. 1993.

[39] H. Schneiderman and M. Nashman. “A Discriminating Feature Tracker for Vision-Based Autonomous Driving”. IEEE Transactions on Robotics and Automation. Vol. 10. NO 6. December 1994.

[40] H. Schneiderman, A. J. Wavering, M. Nashman and R. Lumia. “Real time model-based visual tracking”. International Workshop on Intelligent Robotic Systems. 1994.

[41] J. Slotine and W. Li. Applied Nonlinear Control. Ed. Prentice Hall. 1991.

[42] M. A. Sotelo, A. Zelinsky, F. J. Rodríguez and L. M. Bergasa. “Real-time Road Tracking using Templates Matching”. Third International ICCS Symposia on Intelligent Automation (IIA’99) and Soft Computing (SOCO’99). Genova, Italia. 1999.

[43] M. A. Sotelo, S. Alcalde, J. Reviejo, J. E. Naranjo, R. García, T. de Pedro, C. González. “Vehicle Fuzzy Driving based on DGPS and Vision”. 9th IFSA World Congress and 20th NAFIPS International Conference. July 2001.

[44] A. Stentz and M. Hebert. “A Complete Navigation System for Goal Acquisition in Unknown Environments”. Autonomous Robots, 2, 127-145. Kluwer Academic. 1995.

[45] A. Stentz, M. Hebert and M. Thorpe. *Intelligent Unmanned Ground Vehicles. Autonomous Navigation Research at Carnegie Mellon*. Kluwer Academic Publishers. 1997.

[46] C. Thorpe. *Vision and Navigation: The Carnegie Mellon Navlab*. Kluwer Academic Publishers: Boston. 1990.

[47] D. F. Wolfram Burgard and S. Thrun. “Active Markov Localization for Mobile Robots”. Elsevier Preprint. November 1998.

[48] L. Zhao and C. Thorpe. “Stereo- and Neural Network-based Pedestrian Detection”, in Procs. IEEE Intl. Conf. on Intelligent Transportation Systems’99, (Tokyo, Japan), pp. 298-303, Oct. 1999.