INTERNATIONAL JOURNAL OF CONTROL
https://doi.org/10.1080/00207179.2018.1562225

Taylor & Francis
Taylor & Francis Group

An approach integrating planning and image-based visual servo control for road following and moving obstacles avoidance

Ramses Reyes and Rafael Murrieta-Cid

Centro de Investigación en Matemáticas, Guanajuato, México

ABSTRACT

This paper proposes an approach that integrates planning and image based visual servo control for road following and moving obstacle avoidance. One main objective of this article is to represent a robot's general plan or strategy in the form of a finite state machine or automaton. This automaton is designed previously to execution of the plan and then it is used for any instance of the robot's task. The visual servo control is used to regulate the robot's velocity according to the visual target (task specification) which depends on the state in the automaton. All the algorithms and control laws have been implemented and simulation results and experiments with a real scale-size car-like robot are presented and discussed.

ARTICLE HISTORY
Received 25 June 2018
Accepted 20 October 2018

KEYWORDS
Automaton; planning; visual servo control; car-like robot; road following; obstacles avoidance

1. Introduction

This paper proposes an approach that integrates planning and image based visual servo control for road following and moving obstacle avoidance. Motion planning (Latombe, 1991; Laumond, 1998; LaValle, 2006) and visual servo control (Chaumette & Hutchinson, 2006; Espiau, Chaumette, & Rives, 1992; Fontanelli, Danesi, Belo, Salaris, & Bicchi, 2009; Hutchinson, Hager, & Corke, 1996) are two approaches to command motion systems originated from robotics research which are used very often. Each one of them has advantages and drawbacks. Visual servo control is based on feedback information obtained with sensors specially cameras, which makes the method robust to the presence of uncertainty and reactive. However, visual servoing by its own is not well suited to deal with obstacles that generate both motion and visibility constraints. In contrast, motion planning methods are well suited to deal with obstacles and allow to make deliberative plans. However, typically those methods do not fully take advantage of feedback information provided by sensors. Furthermore, these two paradigms are rarely integrated together to solve more complex tasks. In this work, we propose an approach integrating both of them and exploiting and complementing their advantages. One main goal of this work is to represent the general plan or strategy of the robot to follow the road and pass or avoid collision with other cars in the form of a finite state machine or automaton. This automaton is designed previously to execution and then it is used for any instance of overtaking or collision avoidance. The visual servo control is used to regulate the angular robot velocities according to the visual target (task specification) which depends on the state in the automaton.

1.1 Related work

This work is related to vehicle guidance in robotics, whose goal is to achieve automatic car driving (Buehler, Lagnemma, & Singh, 2008; Nunes, Laugier, & Trivedi, 2009; Thrun et al., 2006). It is also related to the task of visual navigation of mobile robots (Bonin-Font, Ortiz, & Oliver, 2008; Cherubini & Chaumette, 2012; Courbon, Mezouar, & Martinet, 2009; Ohya, Kosaka, & Kak, 1998; SwainOropeza, Devy, & Hutchinson, 2001) and specifically to controlling wheeled robots with visual servoing (Abdelkader, Mezouar, Andreff, & Martinet, 2006; Becerra, López-Nicolás, & Sag"ués, 2011; Cherubini, Chaumette, & Oriolo, 2011; López-Nicolás et al., 2011; Masutani, Mikawa, Maru, & Miyazaki, 1994; SwainOropeza et al., 2001). The task of avoiding collision with obstacles (Cherubini & Chaumette, 2012; Cherubini, Spinder, & Chaumette, 2014; Fraichard & Kuffner, 2012; Minguez, Lamiraux, & Laumond, 2008) is also related with this work.

In Cherubini et al. (2011), the authors present two visual servoing controllers (pose-based and image-based), which allows mobile robots equipped with a fixed camera to reach and follow a continuous path drawn on the ground, a convergence analysis is presented for both controllers. In Cherubini and Chaumette (2012), the authors propose and validate a framework for avoiding obstacles during visual navigation with a wheeled mobile robot. Visual navigation consists of following a path, represented as an ordered set of key images. The robot following that path avoids obstacles which are sensed by an onboard lidar. The camera pan angle is actuated to maintain scene visibility while the robot circumnavigates the obstacle. The risk of collision and the eventual avoiding behaviour are determined

CONTACT Rafael Murrieta-Cid murrieta@cimat.mx, murrietacid@gmail.com Centro de Investigación en Matemáticas, Guanajuato 36023, México

Supplemental data for this article can be accessed here https://doi.org/10.1080/00207179.2018.1562225

using a tentacle-based approach. In Cherubini et al. (2014), the work in Cherubini and Chaumette (2012) is extended to avoid collision with moving obstacles. The proposed approach takes explicitly into account obstacle velocities, estimated using a Kalman-based observer. These velocities are used to predict the obstacle positions. In this work, we also use a laser to detect obstacles, however, in the present work the main goal is integrating planning and image-based visual servo using an automaton, which is not done in the works presented in Cherubini and Chaumette (2012) and Cherubini et al. (2014). Other key differences between the work presented here and the works in Cherubini et al. (2011) and Cherubini and Chaumette (2012) are described in detail below (see Section 1.2).

In this work, we take inspiration from the work presented in Martinez et al. (2018). That work addresses the problem of exploring an unknown, planar, polygonal and simply connected environment. The authors propose an automaton that filters spurious observations to activate feedback-based controllers. The control scheme switches controllers according to observations obtained from the robot's sensor. In this work, we also use an automaton which represents the general plan to accomplish the task; however, the task addressed in this work is different from the task addressed in Martinez et al. (2018). While in Martinez et al. (2018), the goal is to explore an unknown environment, in this paper the tasks are following the road and avoid collision with other moving and static obstacles and overtaking slower cars in a road with two lanes. Furthermore, in Martinez et al. (2018) two different controllers for regulating the linear and angular velocities of the robot are used for each state in the automaton, in contrast in this work the same controllers for regulating the linear and angular velocities are used in all the states in the automaton and only the control set-points are changed. Finally, in Martinez et al. (2018) visual servo control is not used at all.

1.2 Main contributions

There are several approaches for controlling a car using visual servo control, but to our knowledge, the previous works most closely related to our approach are the ones presented in Cherubini and Chaumette (2012); Cherubini et al. (2011). Nevertheless, it is important to stress that while it is true that the visual servo control element in this work is based on the work presented in Cherubini et al. (2011), in this work, we propose a new approach integrating planning and visual servo control to achieve a twofold task: (1) following a road and (2) avoid moving and static obstacles, which correspond for instance to other cars that are eluded or overtaken. The main differences of this work with the works presented in Cherubini and Chaumette (2012); Cherubini et al. (2011) are the following:

(1) To propose a finite state machine or automaton, which corresponds to the general plan or strategy to follow the road and pass or avoid collision with other cars. This plan is done previously to execution and hence it is used for any instance of overtaking or collision avoidance.
(2) In Cherubini and Chaumette (2012), to determine a collision free trajectory in the presence of obstacles, the authors propose to choose a trajectory among a set of options considering the safety of the chosen trajectory. In Cherubini and Chaumette (2012), the angular velocity of the robot is not controlled using visual servoing in the presence of obstacles, the visual servo control only regulates the robot velocity when there are no obstacles perceived by the sensors. In contrast, in the approach proposed in this paper, the robot's angular velocity is always controlled with visual servoing achieving high reactivity to avoid collision with obstacles.
(3) In Cherubini et al. (2011) the authors only considered low robot's velocities, such that the term $\mathrm{d}s/\mathrm{d}t$ can be neglected. This term models the change of position of the visual target in the scene with respect to time, this change occurs since the tracked point moves and it leaves the image and it is replaced with other point. The motion of the tracked point in the image does not totally originate by visual servo control. The linear velocity of the robot is controlled by other means (Cherubini et al., 2011). In contrast, in this work we take into account the term $\mathrm{d}s/\mathrm{d}t$ allowing the robot to move at higher velocities.
(4) In this work we adjust θ^* corresponding to the orientation of the line tangent to the curve modelling the orientation of the road, according to the coordinate x of the tracked point in the image. This strategy allows the control to keep the robot parallel to the road. In Cherubini et al. (2011), the authors maintain θ^* constant and equal to zero.
(5) In this work we use both the curvature of the road and the distance to the obstacles to control the linear velocity of the robot.

2. Problem statement

Given a car-like robot equipped with a camera, which can be represented with a perspective projection model and with an omnidirectional laser range finder with limited range. The objective is to achieve two tasks: (1) follow the road and (2) avoid collision with other moving or static obstacles and passing one single slower car in a two-way road. We assume that at the beginning of the task, the robot lies on the right lane of the road having the road within the camera field of view.

2.1 Robot, camera and image reference frames

To model the vehicle we use three reference frames: A camera reference frame $\mathcal{F}_{\mathcal{C}}(C, X_c, Y_c, Z_c)$, whose origin C is the optical camera centre, an image reference frame $\mathcal{F}_{\mathcal{I}}(O, x, y)$ with origin O at the image centre and a robot reference frame $\mathcal{F}_{\mathcal{R}}(R, X, Y, Z)$, whose origin R is located over the rear wheels' axis at the ground level, see Figure 1. Between the origin C of the camera reference frame and the robot reference frame there is translation t_z along the Z robot's axis and a translation t_y along the Y robot's axis.

3. Automaton

A finite-state machine (FSM) or automaton is defined as a mathematical model of computation, it is conceived as an abstract machine that can be in one of a finite number of states

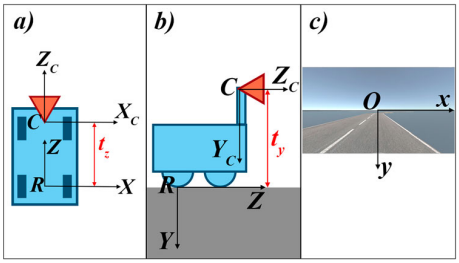


Figure 1. Reference frames. (a) Superior view. (b) Lateral view. (c) Image taken by the camera.

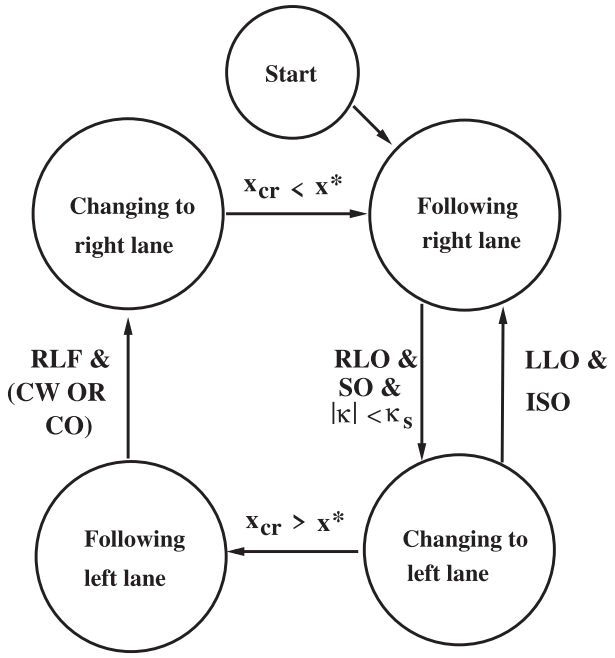


Figure 2. Automaton.

(Hopcroft, Motwani, & Ullman, 2000). Figure 2 shows a graphical representation of an automaton corresponding to the general plan or strategy to follow the road and pass or avoid collision with other cars. The automaton has five states: *Start*, *Following left lane*, *Following right lane*, *Changing to left lane* and *Changing to right lane*. The transitions among the states are represented with arrows, each arrow has a label corresponding to the observation (or condition) required to given a state change to other state. A state corresponds to the general internal condition of the robot, in each one of these states two variables are controlled: (1) the position in the image of the centre of the lane that the robot follows, which is regulated through the angular velocity of the vehicle and (2) the linear velocity of the vehicle, which depends on the presence or absence of obstacles and the curvature of the road. In our current implementation, the state *Start* is trivial, the automaton immediately transits to state *Following right line*, since we assume that the car starts on the right lane. It would be possible to extend this state to reach the road starting with the car placed outside the track.

Sensor measurements obtained with the camera and the laser ranger finder are used in two different ways: (1) They are used as feedback information in controllers of the angular and linear velocities. (2) They are used to answer binary questions, which allow one to change or not from a state to another, we call those elemental decisions 'observations'.

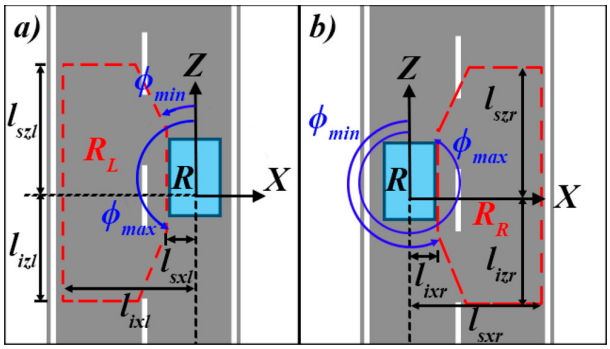


Figure 3. The regions R_L and R_R are bounded by the polygons drawn with dashed lines. (a) Left region (R_L). (b) Right region (R_R).

3.1 Regions

A laser range finder is used to locate the obstacles on a local reference frame attached to the robot. This sensor measures obstacles points in polar coordinates (d, ϕ), where d is the distance from the robot to the obstacle and ϕ is the angle measured in the counterclockwise sense with respect to the Z axis of the robot (see Figure 1(a)) These measurements are used to establish regions around the robot which help to avoid obstacles and to pass slow cars, the regions are defined below.

3.1.1 Left and right regions

From any laser point $p_k(d_k, \phi_k)$ with angle ϕ_k detected within $[\phi_{min}, \phi_{max}]$, the point is represented in cartesian coordinates (X_k, Z_k). If the coordinate Z of a point is larger than an inferior threshold l_{izl} and it is smaller than superior threshold l_{szl}, and the coordinate X of the point is larger than an inferior threshold l_{ixl} and smaller than superior threshold l_{sxl} then the point p_k belongs to left region R_L, that is:

$$p_k \in R_L \quad if\ \phi_k \in [\phi_{min}, \phi_{max}]\ \&\ X_k \in [l_{ixl}, l_{sxl}]\ \&\ Z_k \in [l_{izl}, l_{szl}]$$

The right region R_R is the symmetric case (see Figure 3) and it is defined as follows:

$$p_k \in R_R \quad if\ \phi_k \in [\phi_{min}, \phi_{max}]\ \&\ X_k \in [l_{ixr}, l_{sxr}]\ \&\ Z_k \in [l_{izr}, l_{szr}]$$

3.1.2 Frontal region (R_F), collision region (R_C) and overtaking region (R_O)

The obstacle point $p_k(d, \phi)$ obtained with the laser is given in Cartesian coordinates (X_k, Z_k). If the coordinate Z of a point is larger than an inferior threshold l_{izf} and it is smaller than superior threshold l_{szf}, and the coordinate X of the point is larger than an inferior threshold l_{ixf} and smaller than superior threshold l_{sxf} then the point p_k belongs to the frontal region R_F,

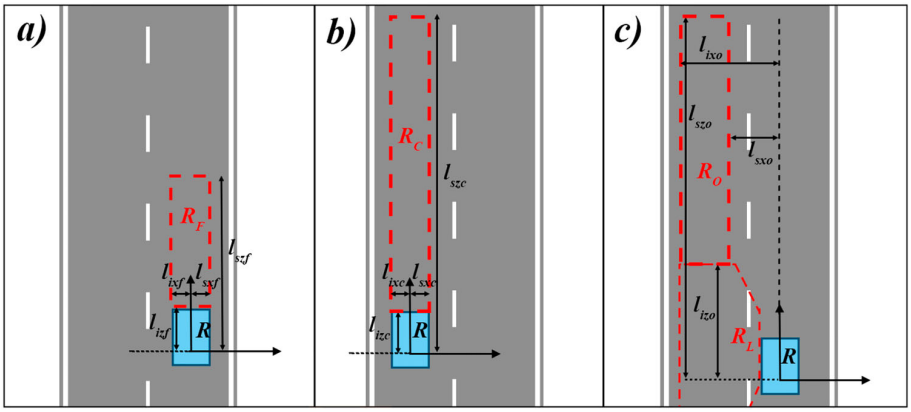


Figure 4. Regions R_F, R_C and R_R are shown with the polygons drawn with dashed lines. (a) Frontal region (R_F). (b) Collision region (R_C). (c) Overtaking region (R_O), left region R_L is also shown in the figure since it is used as a reference, the region R_O starts where region R_L finishes, they share a border.

that is:

$$p_k \in R_F \quad if\ X_k \in [l_{ixf}, l_{sxf}]\ \&\ Z_k \in [l_{izf}, l_{szf}]$$

In an analogous manner, regions R_C and R_O are defined as follows.

$$p_k \in R_C \quad if\ X_k \in [l_{ixc}, l_{sxc}]\ \&\ Z_k \in [l_{izc}, l_{szc}]$$
$$p_k \in R_R \quad if\ X_k \in [l_{ixo}, l_{sxo}]\ \&\ Z_k \in [l_{izo}, l_{szo}]$$

The thresholds l_{szc} and l_{szo} of regions R_C and R_O, respectively, are larger than l_{szf} of region R_F. Regions R_F and R_O only have relevance when the vehicle is in the right lane. In a similar way, region R_C is relevant only when the vehicle lies in the left lane (seeFigure 4).

3.2 Observations

In this work, the conditions triggering changes between states in the automation are called observations. We call them observations since these conditions are stabilised directly from sensors' measurements. The observations correspond to the answer: yes or not of binary questions, most of the observations are deduced from the laser range finder measurements, since they are related to the location of the obstacles with respect to the robot.

The nine observations establishing the transitions between states in the automaton are the following:

- RLO: Right lane occupied. If the laser sensor detects an obstacle within region R_F then this observation is true, otherwise it is false.
- LLO: Left lane occupied. If the laser sensor detects an obstacle within regions R_L or R_O then this observation is true, otherwise it is false.
- RLF: Right lane free. If the laser sensor does not detect an obstacle within regions R_R then this observation is true, otherwise it is false.
- CW: Collision warning. If the laser sensor detects an obstacle within regions R_C then this observation is true, otherwise it is false.

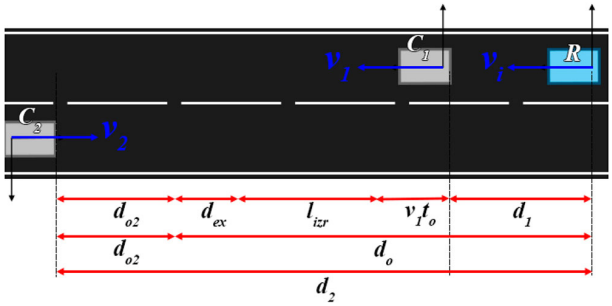


Figure 5. Variables needed to decide whether or not it is safe to overtake a car.

- CO: Car overtaken. If the robot is following the left lane and the car in the right lane is totally overtaken then this observation is true.
- $x_{cr} > x^*$: x_{cr} is the coordinate x in the image of the centre of the road. If x_{cr} is larger than the desired coordinate x^* then this observation is true. This observation indicates that the left lane has been reached.
- $x_{cr} < x^*$: This case is symmetric to the previous one. The desired coordinate x^* can be different depending on the state in the automaton (see Section 5).
- $|\kappa| < \kappa_s$: κ is the curvature of the road, which is obtained using an image processing algorithm (see Section 4.1). If the curvature of the road is smaller than a given threshold κ_s then this observation is true. This is a necessary condition for the robot to overtake a car.
- SO: Safe overtaking. If the laser sensor does not detect an obstacle within regions R_L and R_O then this observation is true, otherwise a procedure to decide whether or not there is enough time to overtake a car is executed. This procedure takes into account the relative[1] velocities of the robot and the other cars (the car to be overtaken and the car moving in opposite direction in the left lane), the distance between the robot and the car to be overtaken, the distance between the robot and the car moving in the opposite direction in the left lane and the maximum robot acceleration. If the procedure determines that there is enough time then the observation is

true otherwise it is false. This procedure is described in detail in the next section.

3.3 Deciding a safe car overtaking

The variables used in the procedure to decide whether it is safe to overtake a car are depicted in Figure 5. In that figure the robot is denoted with an R, the car to be overtake is called C_1, and the car moving in the opposite direction in the left lane is denoted C_2, v_1 and v_2 are the velocities in a global reference frame of car C_1 and car C_2 respectively. Those velocities can be estimated from measurement distances obtained with the laser before making this decision. The distance from the robot to the car C_1 is called d_1 and the distance from the robot to the car C_2 is called d_2.

The distance that the robot travels during the overtaking is called d_o, this distance is composed of four terms: (1) distance d_1, (2) the distance that car C_1 will travel during the time that takes the overtaking called t_o, this distance is equal to $v_1 t_o$, (3) the distance that R must leave behind to the car C_1 to come back safely to the right lane, this distance is considered equal to the inferior limit over the Z axis of the right region R_R called l_{izr} (see Section 3.1) and (4) the distance that the robot will travel over the Z axis while it comes back to the right lane, this distance is called d_{ex}, we assume that the robot will travel this distance while moving to maximum velocity v_{max}.

On the other hand, the distance that car C_2 will travel is called d_{o2}, this distance d_{o2} is equal to the difference $d_2 - d_o$.

To determine whether or not it is safe to overtake car C_1 two times are considered, time t_o that is the time that the robot needs to travel distance d_o using its maximum acceleration and time t_{o2} that is the time in which car C_2 travels distance d_{o2}, it is assumed that car C_2 will travel distance d_{o2} at constant velocity v_2, which is the velocity estimated immediately before to make the decision. If $t_o < t_{o2}$ then there is enough time to overtake car C_1 without colliding with car C_2 and hence observation SO is declared true.

$$\text{SO} = \begin{cases} TRUE & if\ t_o < t_{o2} \\ FALSE & if\ t_o \geq t_{o2} \end{cases} \tag{1}$$

To calculate time t_o we use the following equation:

$$v = \int a(t) \cdot dt + v_i = a_{max} t + v_i \tag{2}$$

we consider the robot acceleration $a(t)$ as the constant a_{max} and v_i is the velocity of the robot at the moment of making the decision.

The distance that the robot travels is given by

$$\begin{aligned} d &= \int v(t) \cdot dt = \int (a_{max} t + v_i) \cdot dt \\ &= a_{max} \int t \cdot dt + v_i \int dt = \frac{a_{max}}{2} t^2 + v_i t \end{aligned} \tag{3}$$

Assuming that the robot will travel distance d_o in time t_o, we have:

$$\frac{a_{max}}{2} t_o^2 + v_i t_o - d_o = 0 \tag{4}$$

Replacing $d_o = d_1 + v_1 t_o + l_{izr} + d_{ex}$ in Equation (4) one gets:

$$\begin{aligned} &\frac{a_{max}}{2} t_o^2 + v_i t_o - (d_1 + v_1 t_o + l_{izr} + d_{ex}) \\ &\quad = \frac{a_{max}}{2} t_o^2 + (v_i - v_1) t_o - d_1 - l_{izr} - d_{ex} \end{aligned} \tag{5}$$

Using the general formula to solve second degree algebraic equations, one gets:

$$t_o = \frac{-(v_i - v_1) \pm \sqrt{(v_i - v_1)^2 - 2(a_{max})(-d_1 - l_{izr} - d_{ex})}}{a_{max}} \tag{6}$$

Finding time t_{o2} is direct, as mentioned above $d_{o2} = d_2 - d_r$ and being v_2 the velocity of car C_2 at the moment of making the decision, one gets:

$$t_{o2} = \frac{d_{o2}}{v_2} \tag{7}$$

Refer to Figure 2, in the automaton, the state *Changing to left lane* is connected to the state *Following to left lane* and also to the state *Following right lane*. This means that at every control cycle the observation SO is evaluated if SO = false and LLO = true, then the action can be aborted without completing it. We allow this behaviour since due to noise measurements (distances or velocities) the calculation of the times t_o and t_{o2} can vary, hence it makes sense to measure the same distances and velocities at every control cycle in order to get a safer overtaking. Furthermore, when the automaton is already in state *Following left lane* it is allowed to transit to state *Changing to right lane* provided that the right lane is free, that is, observation RLF = true, regardless the overtaken is complete, that is CO = true or not.

4. General control scheme

In the section, we shall describe the proposed control scheme, we start describing the image features used.

$$\mathbf{L_s} = \begin{bmatrix} -\frac{1}{Z_C} & 0 & \frac{x}{Z_C} & xy & -(1+x^2) & y \\ \frac{C^2\theta}{t_y} & \frac{S\theta C\theta}{t_y} & -\frac{(xC\theta + yS\theta)C\theta}{t_y} & -(xC\theta + yS\theta)C\theta & -(xC\theta + yS\theta)S\theta & -1 \end{bmatrix} \tag{8}$$

4.1 Image features

To detect the road, we use the algorithm proposed in Udacity online (2018). In that algorithm, first the image is rectified applying a projective perspective transformation (Open, 2018) to get the lines delimiting the road parallel. Second a binarisation procedure is used to obtain the points belonging to the borders of the road. These points (at least 3) are used to obtain a quadratic equation modelling the road borders and it is also used to track the borders in the next images of the sequence. To obtain the curve modelling the centre of a lane both the right and left borders of the lane are used and averaged. Thus, depending on the state in the automaton (see Section 5), a curve

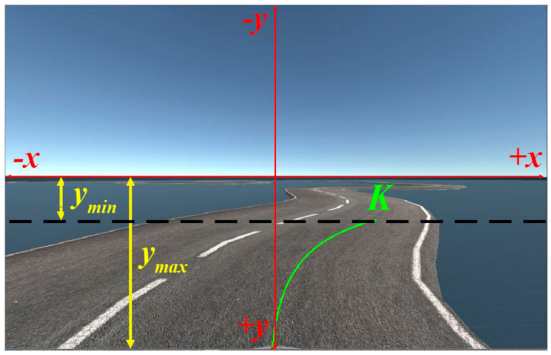


Figure 6. One wishes to define the curve K between y_{min} and y_{max}.

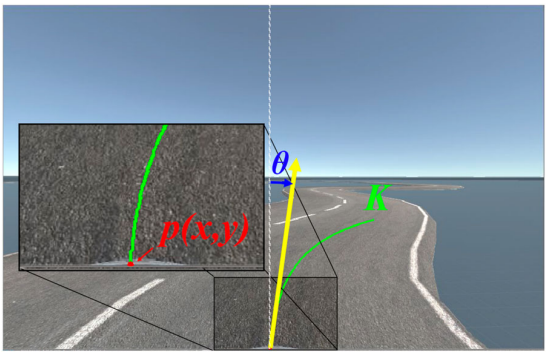


Figure 7. The curve K (starting at point $p(x,y)$) models the centre of the lane. The features x,y and θ are obtained from this representation.

K is used to model the centre of a lane, or the centre of the road when the robot changes lanes to overtake a car. This curve K is defined using the following parametric equation:

$$K(t) = \begin{bmatrix} x_K(t) \\ y_K(t) \end{bmatrix} = \begin{bmatrix} at^2 + bt + c \\ t \end{bmatrix} \tag{9}$$

$x_K(t)$ and $y_K(t)$ are the coordinates on an image reference frame $\mathcal{F}_\mathcal{I}$ and $t \in [y_{min}, y_{max}]$ is the parameter of the curve, which varies from y_{min} to y_{max}, that is the minimum and maximum y value where one expects to find the road, see Figure 6.

From the curve K one gets the following vector of features:

$$\mathbf{f} = \begin{bmatrix} x \\ y \\ \theta \\ \kappa \end{bmatrix} \tag{10}$$

x and y are the coordinates of the point $p(x,y)$, on an image reference frame $\mathcal{F}_\mathcal{I}$, point $p(x,y)$ is located at the place where the curve K intersects the inferior border of the image, θ is the orientation of the line tangent to the curve K at point $p(x,y)$, with respect to the vertical axis measured in counterclockwise sense, see Figure 7. The orientation θ is obtained from the next equation:

$$\theta = \arctan\left(\frac{\dot{x_K}}{\dot{y_K}}\right) \tag{11}$$

$\dot{x_K}$ and $\dot{y_K}$ are the derivatives of x_K and y_K with respect to the parameter t. κ is the curvature of K at the point where the curve K intersects the row of the image at distance y_κ from the inferior border of the image. We measure the curvature of the road some distance in front of the car in order to have a short-term prediction about the curvature of the road, y_κ is determined empirically, see Figure 8. The curvature κ of the road is calculated with the following equation:

$$\kappa = \frac{|\frac{\mathrm{d}^2 K}{\mathrm{d}t^2}|}{|\frac{\mathrm{d}K}{\mathrm{d}t}|^2} \tag{12}$$

The curvature κ is used to control the linear velocity of the robot, it does not affect the angular velocity of the car-like robot.

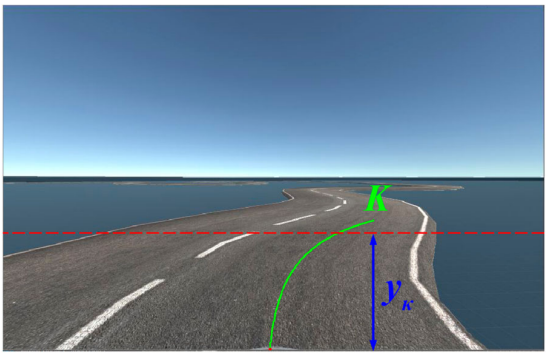


Figure 8. The curvature of the road κ is calculated at a point where the curve K intersects the row of the image at distance y_κ from the inferior border of the image.

4.2 A border of a lane leaves the camera field of view

In our experiments with a real robot (see Section 6.2), we have observed that an extra complication might appear. This complication consists in that one of the borders of a lane leaves the camera field of view.

To estimate the curve modelling a lane's border lying without the camera field of view, one starts from the other border, which is visible. Based on the curve K of the visible border, at least 3 points are selected over the curve K. A perspective projection is applied to these three points to find their locations in a superior perspective view, see Figure 9. With these 3 points one gets a curve K_t lying over the transformed image (superior perspective view). Then one computes the vectors normal to the curve K_t over each point p and points q are placed over the direction of the normal vectors to a distance equivalent to the lane's width in pixels. Using the set of points p a new curve is computed (curve formed by points from q_1 to q_n, in Figure 9). Finally, an inverse perspective projection is applied to the rectified image to find the curve K in the original image that is not visible by the camera.

4.3 Control of linear velocity

The linear velocity of the robot is defined as follows:

$$v^* = v_{t-1} + a \cdot \Delta t \tag{13}$$

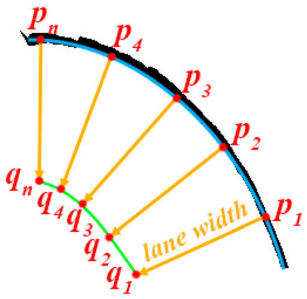


Figure 9. A method to find an approximation to the border of a lane lying without the camera field of view.

v_{t-1} is the linear velocity computed in the previous iteration in the loop and Δt is the time interval between iterations.

The acceleration of the robot is defined as follows:

$$a = \dot{v} = \begin{cases} a_{max} & if\ -\lambda_a(v - v_{ref}) > a_{max} \\ -\lambda_a(v - v_{ref}) & if\ -d_{max} < -\lambda_a(v - v_{ref}) < a_{max} \\ -d_{max} & if\ -\lambda_a(v - v_{ref}) < -d_{max} \end{cases} \tag{14}$$

λ_a is a control gain, v is the current velocity of the robot, v_{ref} is the reference velocity at which one wants that the robot moves, it is different in each state of the automaton and they depend only on the curvature of the road and the distance to the obstacles, they are not regulated using visual servo control, see Section 5. Finally, a_{max} and d_{max} are the maximum acceleration and deceleration of the car-like robot, which depend on the characteristics of the robot's engine.

4.4 Controlling the angular velocity with visual servoing

We start defining $\mathbf{s}$ that is the specifications' vector of the task, which corresponds to follow the road, $\mathbf{s}$ contains the image features x and θ defined in Section 4.1.

$$\mathbf{s} = \begin{bmatrix} x \\ \theta \end{bmatrix} \tag{15}$$

This control approach uses the orientation θ of the line tangent to the curve K modelling the road, this provides to the system a predictability element, that is, it allows to know not only the position of the robot on the road but also it allows to know the location where the car-like robot will be in the near future.

The evolution of specifications of the task is given by the following equation (Chaumette & Hutchinson, 2006; Cherubini et al., 2011):

$$\dot{\mathbf{s}} = \mathbf{L_s}\,{}^{\mathbf{C}}\mathbf{T_R}\mathbf{u} \tag{16}$$

$\mathbf{L_s}$ is the interaction matrix defined by Equation (8), $C\theta$ and $S\theta$ mean $\cos\theta$ and $\sin\theta$, respectively, and Z_C is the depth of the observed point in the world on the camera reference frame. It is well known that image-based visual servo control works correctly even if Z_C is not known precisely (Chaumette & Hutchinson, 2006). The first row of $\mathbf{L_s}$ relates the position of a point in the world on the camera reference frame with its corresponding position over the x axis in the image. The second row relates to the orientation of the line tangent to the curve K in the world with respect to the Z axis on the camera reference frame with the orientation of the same line tangent to the curve K projected on the image, with respect to the y axis of image reference frame $\mathcal{F}_{\mathcal{I}}(O, x, y)$. The inclination of the camera with respect to the X axis, called in Cherubini et al. (2011) ρ, is zero in our robot, hence it is not considered.

${}^{\mathbf{C}}\mathbf{T_R}$ is the homogenous transformation matrix between the robot reference frame $\mathcal{F}_{\mathcal{R}}(R, X, Y, Z)$ and the camera reference frame $\mathcal{F}_{\mathcal{C}}(C, X_c, Y_c, Z_c)$

$${}^{\mathbf{C}}\mathbf{T_R} = \begin{bmatrix} 0 & -t_z \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \tag{17}$$

The vector of control inputs $\mathbf{u}$ of the car-like robot has two elements, the linear v and angular ω velocity.

$$\mathbf{u} = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{18}$$

Multiplying the interaction matrix $\mathbf{L_s}$ by the first column of the transformation matrix ${}^{\mathbf{C}}\mathbf{T_R}$, one gets a Jacobian matrix called $\mathbf{J_v}$, which is related to the linear velocity v of the robot.

$$\mathbf{J_v} = \begin{bmatrix} \frac{x}{Z_C} \\ -\frac{(xC\theta + yS\theta)C\theta}{t_y} \end{bmatrix} \tag{19}$$

Analogously, multiplying the interaction matrix $\mathbf{L_s}$ by the second column of the transformation matrix ${}^{\mathbf{C}}\mathbf{T_R}$, one gets a Jacobian matrix called $\mathbf{J}_\omega$, which is related to the angular velocity ω of the robot.

$$\mathbf{J}_\omega = \begin{bmatrix} \frac{t_z}{Z_C} + 1 + x^2 \\ -\frac{t_z C^2\theta}{t_y} + (xC\theta + yS\theta)S\theta \end{bmatrix} \tag{20}$$

However, Equation (16) is only applicable when the target observed point in the world does not move or it moves

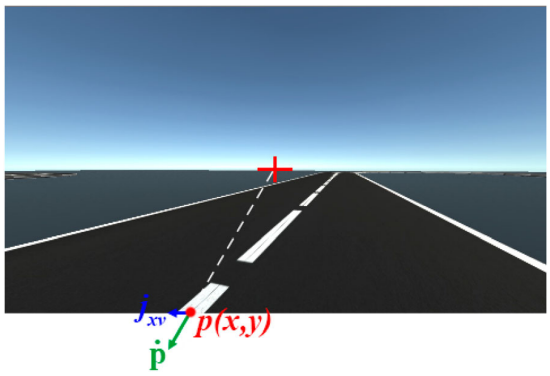


Figure 10. Typically the IBVS tracks points fixed in the world, if that was the case, then a movement of the robot forward in the straight line will produce that the point p moves over the vector $\dot{\mathbf{p}}$.

slowly, such that the motion can be neglected as in Cherubini et al. (2011). But in this work the target point in the world changes all the time. If the same point is observed then it will leave the camera field of view. Thus, to consider the motion of the observed target point a factor must be added to Equation (16), yielding the next equation

$$\dot{\mathbf{s}} = \mathbf{L_s}\,^{\mathbf{C}}\mathbf{T_R}\mathbf{u} + \frac{\mathrm{d}\mathbf{s}}{\mathrm{d}t} \tag{21}$$

The factor $\mathrm{d}/\mathbf{s}\mathrm{d}t$ corresponds to the motion of the observed target point generated by factors not controlled by the visual servoing such as the linear robot velocity v. The linear velocity v generates undesired motions that must be compensated by the angular velocity w.

To better understand the reason why it is important to consider the factor $\mathrm{d}\mathbf{s}/\mathrm{d}t$, see Figure 10, if the robot moves forward in a straight line, the angular velocity is zero, then the points in the image will move in the direction of the centre of the image but in appositive sense, leaving the image. Thus, the point p in the image will move over the vector $\dot{\mathbf{p}}$. One can see that such a vector has a component over the x axis of the image, which corresponds to $j_{xv}v^*$, j_{xv}being the first element of the Jacobian matrix $\mathbf{J_v}$, that is x/Z_C. To compare this motion and keep the point p in the same x coordinate, the vehicle must turn to the left in spite of the car-like robot being already aligned with the road.

Nevertheless in this work, the point p continuously changes from an image to another, to cancel the motion of point p the term $\mathrm{d}\mathbf{s}/\mathrm{d}t$ must be the negative of the motion of p generated by the linear velocity v^*, since that velocity is not controlled with visual servoing. Thus, the term $\mathrm{d}\mathbf{s}/\mathrm{d}t$ is given by the following equation:

$$\frac{\mathrm{d}\mathbf{s}}{\mathrm{d}t} = -\mathbf{J_v}v^* \tag{22}$$

The vector of error $\mathbf{e}$ is defined as

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* \tag{23}$$

The reference desired features $\mathbf{s}^*$ will be defined in the next section, they are not always constant. The evolution of the error over time is obtained deriving the previous equation and it is given by

$$\dot{\mathbf{e}} = \mathbf{L_s}\,^{\mathbf{C}}\mathbf{T_R}\mathbf{u} + \frac{\mathrm{d}\mathbf{s}}{\mathrm{d}t} - \dot{\mathbf{s}}^* \tag{24}$$

Using the Jacobian matrices (19) and (20) and substituting (22), Equation (24) can be written as

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = \mathbf{J_v}v + \mathbf{J}_\omega\omega - \mathbf{J_v}v - \dot{\mathbf{s}}^* \tag{25}$$

Which is equivalent to:

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = \mathbf{J}_\omega\omega - \dot{\mathbf{s}}^* \tag{26}$$

Let Λ be a diagonal matrix of gains:

$$\Lambda = \begin{bmatrix} \lambda_x & 0 \\ 0 & \lambda_\theta \end{bmatrix} \tag{27}$$

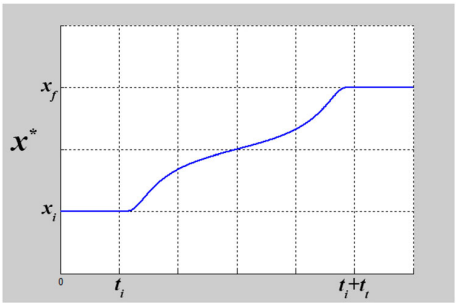


Figure 11. Graphic of function x^*, initial x_i measured at time t_i and final desired value x_f which is reached in predetermined time t_t.

One wants that the error obeys the following dynamic:

$$\dot{\mathbf{e}} = -\Lambda\mathbf{e} \tag{28}$$

Finally, the control law is given by the following equation:

$$\omega = -\mathbf{J}_\omega^+(\Lambda\mathbf{e} - \dot{\mathbf{s}}^*) \tag{29}$$

The symbol + as superindex denotes the Moore-Penrose Pseudoinverse of the matrix $\mathbf{J}_\omega$.

4.5 Changing the task specifications s*

Since from a state to another the position of the observed point p might change abruptly, then the error in the features in the image might also change from a small to a large value. This will generate undesired jumps in the controls, specifically in the angular velocity, which is controlled with visual servoing. In order to avoid this discontinuities in the errors, we will enforce that the desired x^* changes smoothly from an initial x_i to final x_f value in a predetermined time according to the next equation.

$$x^* = \begin{cases} x_i + (x_f + x_i)\left\{1 - \frac{1}{2}\left[1 + \tanh\left(\frac{1}{t-t_i} + \frac{1}{t-t_i-t_t}\right)\right]\right\} \\ \qquad \text{if } t \leq t_i + t_t \\ x_f \quad \text{if } t > t_i + t_t \end{cases} \tag{30}$$

t is the elapsed time, t_i is the initial time when the system enters a new state in the automaton, and t_t is the time that it takes to change from an initial x_i to final x_f value. Figure 11 shows the behaviour of the function.

The desired orientation of the line tangent to the curve K is the one pointing to the image centre, which shows that the robot is aligned with the road, in contrast with the work presented in Cherubini et al. (2011), where it is kept constant and equal to zero.

Thus, θ^* is given by the next equation:

$$\theta^* = \arctan\frac{x}{y} \tag{31}$$

In Equation (29) defining the control law for the angular velocity ω, it considers the term $\dot{\mathbf{s}}^*$ and not $\mathbf{s}^*$, $\dot{\mathbf{s}}^*$ corresponds to the derivatives of the task specifications.

$\dot{x^*}$ is given by the following equation:

$$\dot{x^*} = \begin{cases} \frac{1}{2}(\dot{x_f} + \dot{x_i}) \left[\frac{\frac{1}{(t-t_i)^2} + \frac{1}{(t-t_i-t_t)^2}}{\cosh^2\left(\frac{1}{t-t_i} + \frac{1}{t-t_i-t_t}\right)} \right] & \text{if } t \leq t_i + t_t \\ 0 \quad \text{if } t > t_i + t_t \end{cases} \tag{32}$$

The desired derivative of θ^* is given by the following equation:

$$\dot{\theta^*} = \frac{\dot{x}y}{x^2 + y^2} \tag{33}$$

The derivative of x is approximated with finite differences $\dot{x} = (x_k - x_{k-1})/(t_k - t_{k-1})$. Thus, the derivative of vector of task specifications $\mathbf{s}^*$ is given by

$$\dot{\mathbf{s}^*} = \begin{bmatrix} \dot{x^*} \\ \dot{\theta^*} \end{bmatrix} \tag{34}$$

5. Control set-points and target velocities according to the state in the automaton

5.1 Following right lane

In this state $x_f = 0$, since we want that the centre of the right lane be located at the centre of the image.

The linear velocity of the robot given by Equation (13) depends on a reference velocity v_{ref}, which varies according to the state, in this state the linear velocity is denoted v_{refR} and it is defined by the following equation:

$$v_{refR} = [v_{min} + \sigma(v_{nom} - v_{min})]\tau_R \tag{35}$$

v_{min} is the velocity of the robot such that it can travel the curve with the highest curvature, v_{nom} is the nominal linear velocity at which one wants that the robot travels in a straight line without having obstacles close, σ is a function that adapts the linear velocity according to the road curvature κ and τ_R is a function that adapts the linear velocity according to the distance from the robot to the next car in the right lane, called distance d_{nc}. Function $\sigma(\kappa)$ is given by the following equation:

$$\sigma(\kappa) = \begin{cases} 1 - (\frac{\kappa}{\kappa_{max}})^2 & \text{if } |\kappa| < \kappa_{max} \\ 0 & \text{if } |\kappa| \geq \kappa_{max} \end{cases} \tag{36}$$

κ_{max} is the maximum curvature that the robot can travel (see Figure 12), the function τ_R is defined by the following equation:

$$\tau_R(d_{nc}) = \begin{cases} 0 \\ \text{if } d_{nc} < d_{min} \\ 1 - \frac{1}{2}[1 + \tanh(\frac{1}{d_{nc} - d_{min}} + \frac{1}{d_{nc} - l_{szf}})] \\ \text{if } d_{min} < d_{nc} < l_{szf} \\ 1 \\ \text{if } d_{nc} \geq l_{szf} \end{cases} \tag{37}$$

d_{min} is the minimum allowed distance between the robot and the next car in the right lane, when the robot is to d_{min} from the next car in the right lane its linear velocity is zero, l_{szf} is the superior threshold over the Z axis of region R_F (see Figure 4a)). Figure 13 shows the graphic of function 13.

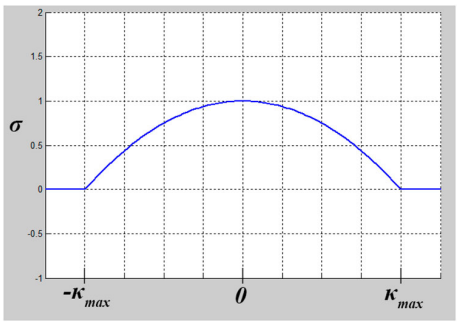


Figure 12. Graphic of function σ, its maximum value corresponds to $\kappa = 0$ and it decreases exponentially until it reaches zero when $\kappa = -\kappa_{max}$ or $\kappa = \kappa_{max}$.

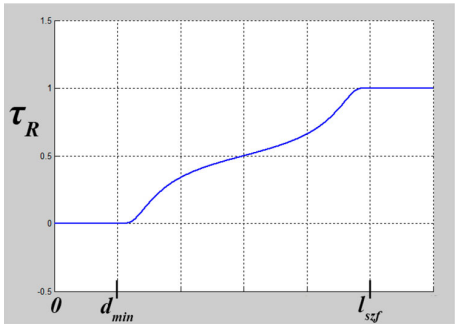


Figure 13. Graphic of function τ_R, it has its maximum value of 1 when $d_{nc} \geq l_{szf}$ and it has its minimum value when $d_{nc} < d_{min}$.

5.2 Changing to left lane

When the robot is in the right lane and the observation to change to the left lane appears the robot still is in the right lane and it is difficult to detect the centre of the left lane, then the robot detects instead the centre of the road, thus in this state the point p corresponds to the centre of the road. The final desired value x_f is a coordinate $x > 0$ such that the left border of the road is within the field of view of the camera and it intersects the inferior border of the image. The reference velocity v_{ref} is the same that in the previous state (*Following right lane*).

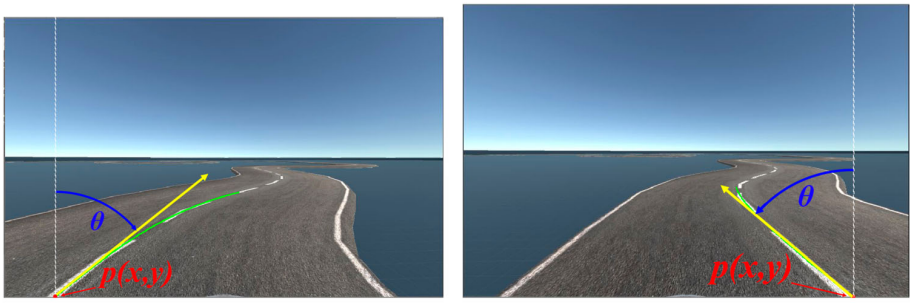
5.3 Following left lane

At the moment the automation transits to this state, the robot is able to sense both borders left and right ones of the left lane, hence, the point p is the centre of the left lane and $x_f = 0$, since the goal is to keep the centre of the left lane at the centre of the image. The reference velocity (v_{refL}) is given by the next equation:

$$v_{refL} = [v_{min} + \sigma(v_{max} - v_{min})] \tag{38}$$

v_{max} is n times the nominal velocity v_{nom} with $n > 1$, such that $v_{max} > v_{nom}$. Function σ is the same that in the state *Following right lane*.

As we said before, if the automaton is in this state the procedure described in Section 3.3 has decided that the overtaking is safe. But if a car moving in the opposite direction is getting too close to the robot (observation CW = true) and there is enough free space in the right lane (observation RLF = true) then the



(a) Changing to left lane (b) Changing to right lane

Figure 14. When the robot changes of lane, the centre of the road is point p and not the centre of a lane.

robot is allowed to come back to the right lane without overtaking the car in the right lane (observation CO = false), see the observations required to transit to state *Changing to right lane* (see Figure 2).

5.4 Changing to right lane

Similar to the state *Changing to left lane* the observed and tracked point p is the centre of the road, specifically in this state, the point p is at the place where the right border of the left lane intersects the inferior border of the image, see Figure 14b). The desired x_f for point p is a coordinate $x < 0$ such that it guarantees that the right lane of the road is within the field of view of the camera.

The reference velocity v_{ref} is the same that the one used in state *Following right lane* (see Section 5.1).

6. Simulations and experiments in a real robot

6.1 Simulation results

All our simulation experiments were run on a 2.3 GHz In- tel Core i3-6100U dual-core processor PC, equipped with 8 GB of RAM, running Windows 10, and were programmed in C# and C++ using the multipurpose game engine Unity.

First, we present simulation experiments without other cars, only the car-like robot is in the track. For each experiment, 3 trials were done with the same parameters. The track has a length of 710 meters, see Figure 15, the car-like robot is 3.4 meters long, 1.8 meters width and 1.3 meters high. The common parameters for this first set of experiments are the following: $d_{max} = 30$, $t_z = 2.3$, $t_y = 1.8$, $L = 3$, $Z_C = 6.5121$, $\lambda_a = 4$, and $\kappa_{max} = 3$. d_{max}

is the maximum car deceleration given in m/sec^2, t_z and t_y are offsets (see Section 2.1), L is the distance between the front and rear wheels axis, Z_C is the assumed distance between the camera and the road-which does not need to be accurate in image based visual servo control-, all these parameters are given in meters. λ_a is a control gain related to the car acceleration and κ_{max} is the maximun road curvature, these last two parameters are adimensional.

In Table 1, we vary the following parameters: λ_x and λ_θ are control gains, related respectively to the x coordinate of tracked point p and the angle of the line tangent to the road

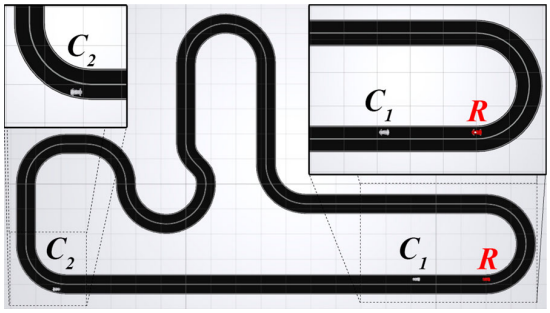


Figure 15. Track used in simulation experiments and position of the vehicles before a risky overtaking.

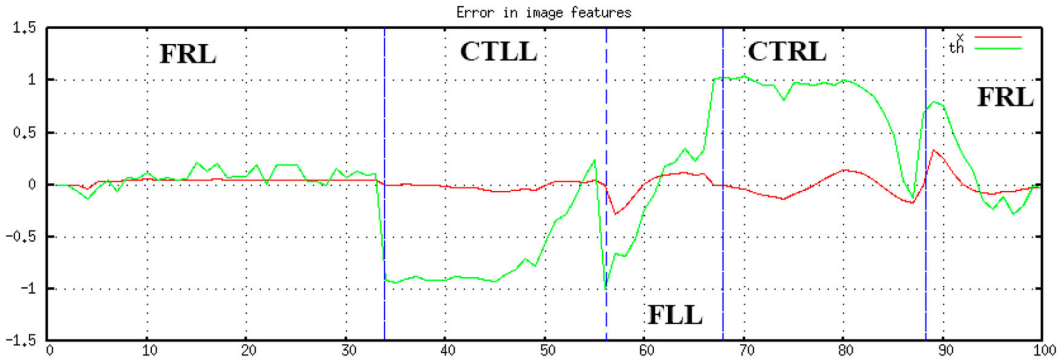
Table 1. Results in simulations without other cars.

E#	λ_x	λ_θ	v_{min}	v_{nom}	a_{max}	SA	t_{av}
1	9	2	10	32	8	3	33.1
2	9	3	10	32	8	0	–
3	9	4	12	32	8	2	30.95
4	9	4	11	32	8	3	31.86
5	9	4	11	32	10	1	31.2
6	9	4	11	32	9	1	31.1
7	10	4	11	31	8	2	33.3
8	10	4	10	30	8	3	33.2
9	10	4	10	30	10	3	32.1
10	10	4	10	31	10	3	31.5
11	10	4	10	32	9	1	31.76

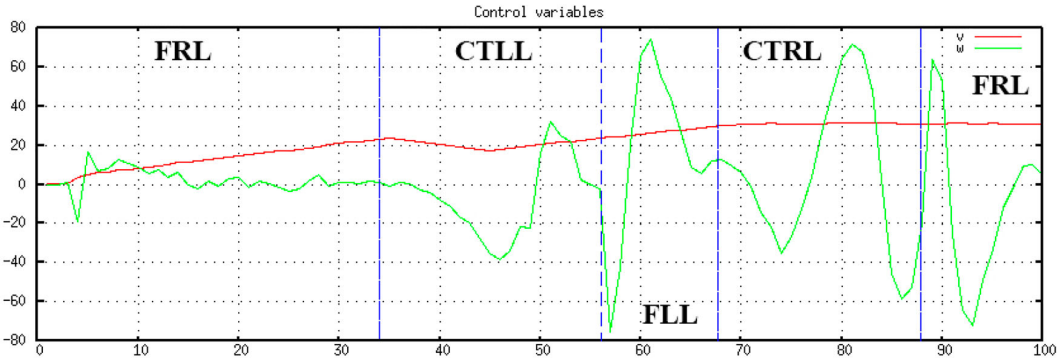
at point p, v_{min} is the minimum car velocity, v_{nom} is the nominal car velocity, a_{max} is the maximum car acceleration. In all the simulations the maximum car-like robot velocity that the vehicle can reach is the double of the nominal velocity and *in all the experiments with the real robot the maximum car-like robot velocity is 1.5 times the nominal velocity*. The main results are the two following: SA is the number of successful attempts, that is the number of times that the robot completes a lap in the circuit, and t_{av} is the average time to travel a single lap in the track.

We conclude that the ability of the car-like robot to complete a lap depends mainly in the control gains λ_x and λ_θ. The time to complete a lap provided a well tuned gains depends on the nominal velocity v_{nom} and the maximun acceleration a_{max}. The larger these quantities the shorter the time.

The second set of simulation experiments was done with two cars besides the robot, to study car overtaking. The two



(a) Error in image features



(b) Control variables ω and v

Figure 16. Graphics of the error in the image features and control variables ω and v in one simulation during a car overtaking.

other cars are not controlled, they follow a predefined path in the track. In this second set of experiments 5 trials were done for each experiment with the same parameters. The common parameters are the following: $Z_C = 6.5121$, $d_{max} = 30$, $\lambda_a = 4$, $\kappa_{max} = 3$, $t_z = 2.3$, $t_y = 1.8$, $L = 3$, $v_{max} = 2 \times v_{nom}$, $\kappa_s = 1.5$, $d_{min} = 4$, $l_{szf} = 20$.

Recall that κ_s is a threshold of the road curvature and l_{szf} is the distance from the origin of the reference frame related to the robot and the superior limit of the region R_F, see Figure 4; d_{min} is the minimum allowed distance between the robot and the car in the same lane in front of the robot, if the distance between the robot an the car is smaller than d_{min} then the robot stops. We set $\lambda_x = 10$ and $\lambda_\theta = 4$, which were the gains that gave the best performance in the previous set of experiments.

Table 2 shows the parameters that we vary, it also shows the 3 main results in this set of experiments. They are: SA the number of successful attempts, t_{av} the average time to travel a single lap in the circuit, and RO that indicates the number of times that a risky overtaking is done. A risky overtaking corresponds to an overtaking in which the robot does not wait for the car in the left lane to pass to overtake the car in the same lane that the car-like robot. The initial positions of the robot and the two cars, when a risky overtaking is done, are shown in Figure 15. Based on these experiments we conclude that a successful risky overtaking depends mainly on the maximun robot acceleration a_{max}, the larger it is the better.

Table 2. Results in simulations with other cars

E#	v_{min}	v_{nom}	a_{max}	SA	RO	t_{av}
1	11	28	8	5	5	33.54
2	11	28	6	4	1	39.22
3	11	30	6	4	0	39.77
4	11	30	8	4	4	32.55
5	11	30	10	5	5	31.52
6	11	32	6	4	0	39.6
7	11	32	7	5	5	32.72
8	11	28	7	5	5	34.56
9	10	26	7	5	5	36.36
10	10	32	10	5	5	31.76

Figure 16(a) shows the evolution of the error in the image features, and Figure 16(b) shows the control variables ω and v, for a simulation with other cars during a car overtaking. The vertical blue lines in the figures indicate the changes of states: Following left lane (FLL), Following right lane (FRL), Changing to right lane (CTRL), Changing to left lane (CTLL) and the error in x is shown with a curve in medium gray and error in theta (th) is shown with a curve in light gray. Sometimes the error in the θ variable changes abruptly, this is due to that the measured θ can change from the one related to the centre of the lane to the one related to the centre of the road. A possibility to make this change smoother is to track the measured θ as we do with the desired x (see Section 4.5). However, that strategy might require too much time to change

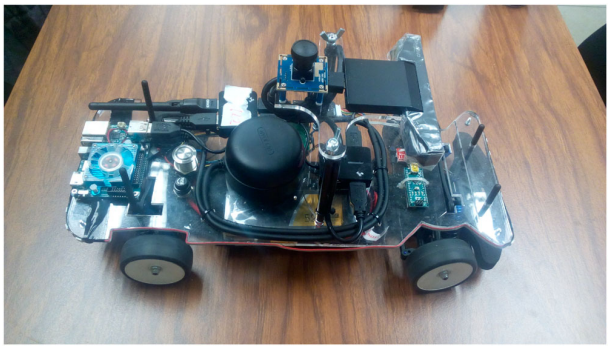


Figure 17. The car-like robot and the sensors.

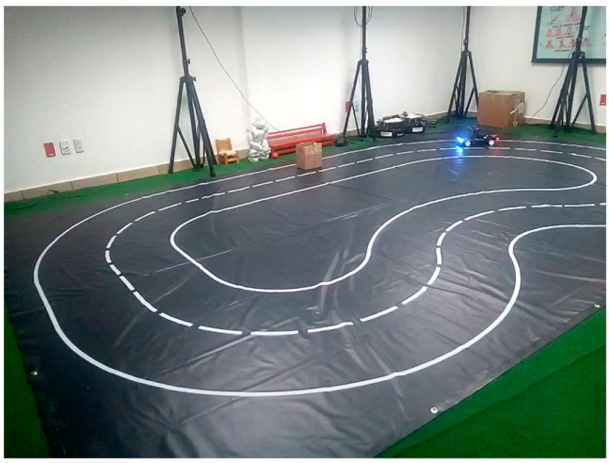


Figure 18. Experiment with a car-like robot avoiding a static obstacle.

the car direction reducing the reactivity to avoid collision with an obstacle.

6.2 Experiments with a real robot

In all the experiments, we have used a car-like robot (size-scale vehicle 1:10) called AutoNOMOS mini, see Figure 17, which has been developed at Freie Universität Berlin AutoModelCarWiki Online (2017). The on-board computer is a card Odroid XU4 with 64GB of RAM, running Linux and Robotic Operating System (ROS). The vehicle is equipped with a laser range finder RPLidar 360, which is used to detect obstacles around the robot and a Kinect type video camera, which is used to detect the lanes in the road. We do not use depth information from the camera, only 2D images.

Each experiment with the real car-like robot was done only once. Figure 18 shows the track used in the experiments, the length of the track in the centre of the exterior lane is approximately 7.65 meters and the width of each lane is 0.4 meters. The common parameters in the experiments are the following: $Z_C = 6.5121$, $\lambda_a = 4$, $\kappa_{max} = 2$, $t_z = 0.2$, $t_y = 0.18$, $L = 0.25$, $v_{max} = 1.5 \times v_{nom}$, $\kappa_s = 1.5$. Table 3 shows the parameters that were varied. The main result is the time t (measured in seconds) to complete a single lap in the circuit. A fail to complete a single lap in the track is indicated with a "-" in the column indicating the time. The nominal robot velocity varies from 3.2 to 3.6 meters per second, which corresponds to a range between 11.52 and 12.96 kilometers per hour. Note that considering that the scale of the vehicle is 1:10 (the radius of vehicle wheels is 10 times smaller than in a real size one), this would correspond to a velocity range between 115 and 129 kilometers per hour in real size vehicle.

Table 3. Results in real experiments without obstacles.

E#	λ_x	λ_θ	v_{min}	v_{nom}	a_{max}	d_{max}	t
1	10	4	1.7	3.2	0.6	2.5	30
2	10	4	1.7	3.2	0.8	2.5	30
3	10	4	1.9	3.4	0.8	2.5	28
4	10	4	1.9	3.6	0.6	2	–
5	10	4	1.9	3.5	0.6	2	–
6	12	4	1.9	3.5	0.6	2	–
7	12	4	1.5	3.5	0.6	2	30
8	12	4	1.5	3.3	0.6	2	30

Table 4. Experiments with one obstacle.

E#	λ_x	λ_θ	v_{min}	v_{nom}	a_{max}	d_{max}	t
1	9	4	1	2	0.6	2.5	42
2	10	5	1.2	2.4	0.6	2.5	37
3	11	5	1.2	2.4	0.4	3	42
4	10	5	1.2	2.6	0.4	3.5	36
5	10	5	1.2	2.8	0.5	3.5	33
6	10	5	1.2	3.2	0.5	3.5	30
7	10	5	1.4	3.2	0.5	3.5	31
8	10	5	1.5	3.2	0.5	5	30
9	10	5	1.5	3.4	0.5	6	28
10	10	5	1.5	3.5	0.5	6	29

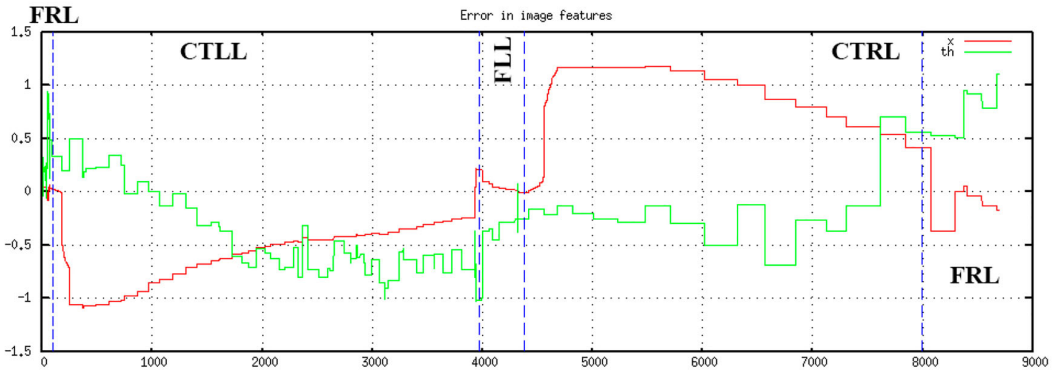
In experiments with a real robot, we conclude that provided well tuned control gains, the ability of the car-like robot to complete a lap depends mainly on the minimum allowed velocity v_{min}.

Figure 18 shows an experiment with the car-like robot, where the robot avoids collision with a static object lying on the track's right lane. The car changes lane to avoid the obstacle and then it gets back to the right lane (see the video showing this experiment in the multi-media material of the paper.[2] In these experiments we have noticed that some times the robot stops when it encounters an obstacle before changing line. That is due to size region R_F, see Figure 4, a small size region was used. A longer region will detect obstacles outside the road producing the car to change lane when it is not required.

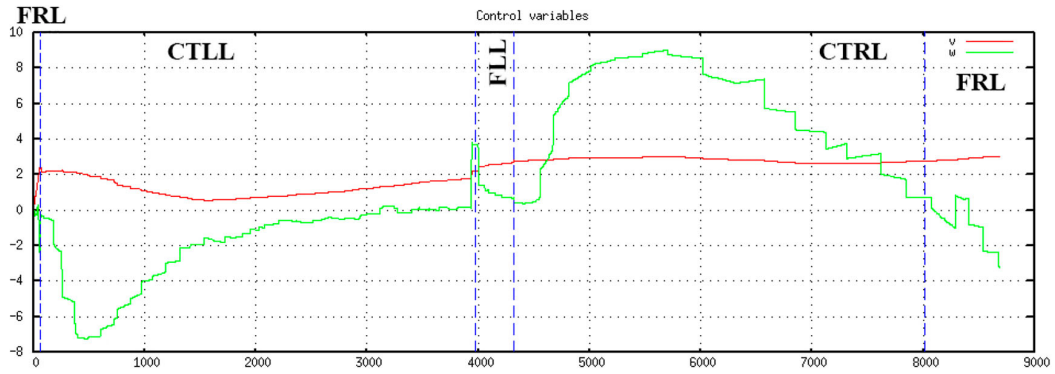
We present now a last set of experiments where the robot avoids a static obstacle. The common parameters in this set of experiments were the same that in the previous set. Table 3 shows the parameters that were varied. Again, the main result is the time t to complete a single lap in the circuit.

In experiments with a real robot in the presence of obstacles, we conclude that provided well tuned controls gains, the time to complete a lap depends mainly on nominal velocity v_{nom}, the larger it is, the shorter the time needed to complete a lap.

Figure 19(a) shows the error in the image features and Figure 19(b) shows the control variables ω and v, for a experiment with a static obstacle during the collision avoidance. The vertical dotted lines in the figures indicate the changes of states: Following left lane (FLL), Following right lane (FRL), Changing to right lane (CTRL) and Changing to left lane (CTLL).



(a) Error in image features



(b) Control variable ω and v

Figure 19. Graphics of the error in the image features and control variables ω and v in an experiment with the real robot and one static obstacle.

We can observe that these graphics are less smooth in comparison with the previous ones resulting from the simulations, we think that it is due to the image processing method which might detect wrong feature points which induces noise in the variable errors.

Unfortunately, we do not have a second car to test experimentally the car overtaking. However, a careful analysis with several simulations was done to assess the proposed method for car overtaking. In the multi-media material, we have included a video, in which two simulations and one experiment in the real robot are presented. The paper multimedia material (a video) is in the following link:
Link to the video: https://figshare.com/s/9deb155fabec4c6ee647

The present work delivers experimental results that addressed the scenario in which the car-like robot moves in a closed circuit, and in which the tasks given to the robot are following the road, avoid collision with other moving and static obstacles, and overtaking slower cars in a road with two lanes. In this context, we present a experimental system composed by a car-like robot equipped with a camera used to detect the lanes' borders, and an omnidirectional laser range finder used to detect moving and static obstacles. The proposed approach integrates an automaton and image-based visual servo control for accomplishing the tasks given to the car-like robot. Two main improvements with respect to previous works are the following:

(1) To the best of our knowledge the previous work most closely related to this work are the ones presented in Cherubini et al. (2011) and Cherubini and Chaumette (2012). The proposed approach is able to decide whether or not is safe to overtake a car in the right lane estimating the time needed to do it, considering the location or absence of a car moving in opposite direction in the left lane, this capability is demonstrated in several simulations which consider realistic values of maximal accelerations and velocity of the cars. The works presented in Cherubini et al. (2011) and Cherubini and Chaumette (2012) do not present this capability.
(2) Another important novelty proposed in this work with respect to methods using image-based visual servo is that the orientation θ^* of the line tangent to the curve modelling the orientation of the road is adjusted, according to the coordinate x of the tracked point in the image. This strategy allows the control to keep the robot parallel to the road. To the best of our knowledge this strategy is not presented in previous works in which the car is controlled

using image-based visual servo. This capability is verified in experiments with a physical robot with good results.

7. Conclusions and future work

In this paper, we have proposed an approach integrating planning and visual servo control to achieve a twofold task: (1) following a road and (2) avoid moving and static obstacles, which correspond for instance to other cars that are eluded or overtaken. We have proposed a finite state machine or automaton, which corresponds to the general plan or strategy to follow the road and pass or avoid collision with other cars. This plan is done previously to execution and then it is used for any instance of overtaking or collision avoidance. In this approach, the robot's angular velocity is always controlled with visual servoing achieving high reactivity to avoid collision with obstacles. We have taken into account the term $\mathrm{d}s/\mathrm{d}t$ modelling the change of position with respect to time of the visual target in the scene allowing the robot to move at higher velocities. The variable θ^* corresponding to the orientation of the line tangent to the curve modelling the orientation of the road is adjusted according to the coordinate x of the tracked point in the image. This strategy allows the control to keep the robot parallel to the road. All the algorithms and control laws have been implemented and simulation results and experiments with a real car-like robot are presented and discussed. Our experiments have empirically shown that the proposed method works correctly in simulations and experiments.

There are several directions for future work, for instance: (1) Overtake more than one single car in the right lane, this would allow to consider more complex situations in which there are several cars moving in the same direction as the car-like robot. In that case, it may be necessary for the car-like robot to determine the time needed to overtake two or more cars in the right lane at once without coming back to the right lane. (2) Detect and discard static obstacles lying outside the road. In our experiments we have noticed that sometimes the robot stops when it encounters an obstacle before changing line. That is due to a small size of region R_F. A longer region will detect obstacles outside the road producing the car to change lane when it is not required. For that reason, it would be useful to detect whether or not an obstacle lies on the road. (3) Currently we are only considering that the car-like robot moves in a closed circuit as a car race track. It would more challenging to consider a road with bifurcations that can be used to model for instance a highway. (4) We think that an important problem is to obtain an automatic driving being comfortable for a human user. We plan to combine virtual reality, machine learning techniques and statistical analysis to find the parameters that produce a comfortable ride for a human user. Using machine learning methods we plan to find the parameters (acceleration, distance to the other cars, etc.) producing different driving styles (e.g. aggressive, conservative, etc.). Using a virtual reality headset we would like to expose human users to the different driving styles. We plan to make polls and statistical analysis to determine how comfortable the experience is for the user, for instance according to the user age or gender. We believe that such research would be very useful to provide an adequate driving style according to the user preferences. (5) Develop methods for ensuring the safety of autonomous vehicles as in the following works: (Decastro et al., 2018; Fisac et al., 2018; Schwarting, Alonso-Mora, Paull, Karaman, & Rus, 2017). This is paramount for successful deployment of autonomous vehicles.

Notes

1. The velocities of the other cars in a global reference frame can be obtained from the relative velocities, knowing the velocity of the car-like robot itself.
2. The paper multimedia material (a video) is in the following link:Link to the video: https://figshare.com/s/9deb155fabec4c6ee647

Acknowledgments

The authors would like to thank Prof. Raul Rojas for providing us with the car-like robot (size-scale vehicle) used in the experiments. The authors would also like to acknowledge the financial support of Intel Corporation for the development of this work.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by Consejo Nacional de Ciencia y Tecnología [220796] and Intel Corporation [CIMAT/INTEL Project].

ORCID

Rafael Murrieta-Cid http://orcid.org/0000-0002-8334-5287

References

Abdelkader, H. H., Mezouar, Y., Andreff, N., & Martinet, P. (2006). *Omnidirectional visual servoing from polar lines*. Paper prsented at the Proceedings of the IEEE International Conference on Robotics and Automation, Orlando.

Becerra, H. M., López-Nicolás, G., & Sag"ués, C. (2011). A sliding-mode-control law for mobile robots based on epipolar visual servoing from three views. *IEEE Transactions on Robotics*, *27*, 175–183.

Bonin-Font, F., Ortiz, A., & Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, *53*, 263–296.

Buehler, M., Lagnemma, K., & Singh, S. (eds) (2008). Special Issue on the 2007 DARPA Urban Challenge, Parts I–III. *Journal of Field Robotics*, *25*(8-10), 423–860.

Chaumette, F., & Hutchinson, S. (2006, December). Visual servo control, part I: Basic approaches. *IEEE Robotics & Automation Magazine*, *13*(4), 82–90.

Cherubini, A., & Chaumette, F. (2012, September). Visual navigation of a mobile robot with laser-based collision avoidance. *The International Journal of Robotics Research*, *32*, 189–205. doi:10.1177/0278364912460413

Cherubini, A., Chaumette, F., & Oriolo, G. (2011, April). Visual servoing for path reaching with nonholonomic robots. *Robotica*, *29*, 1037–1048. doi:10.1017/S0263574711000221

Cherubini, A., Spinder, F., & Chaumette, F. (2014, October). Autonomous visual navigation and laser-based moving obstacle avoidance. *IEEE Transactions on Intelligent Transportation Systems*, *15*(5), 2101–2110.

Courbon, J., Mezouar, Y., & Martinet, P. (2009). Autonomous navigation of vehicles from a visual memory using a generic camera model. *IEEE Transactions on Intelligent Transportation Systems*, *10*, 392–402.

Decastro, J., Liebenwein, L., Vasile, C.-I., Tedrake, R., Karaman, S., & Rus, D. (2018). *Counterexample-guided safety contracts for autonomous driving*. Accepted to be present at the 13th International Workshop on the Algorithmic Foundations of Robotics, Mérida, México.

Espiau, B., Chaumette, F., & Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, *8*(3), 313–326.

Fisac, J., Bajcsy, A., Herbert, S., Fridovich-Keil, D., Wang., S., Tomlin, C., & Dragan, A. (2018). *Probabilistically safe robot planning with confidence-based human predictions*. Proceedings of Robotics: Science and Systems. Pittsburgh, PA.

Fontanelli, D., Danesi, A., Belo, F. A. W, Salaris, P., & Bicchi, A. (2009). Visual servoing in the large. *The International Journal of Robotics Research*, *28*, 802–814.

Fraichard, T., & Kuffner, Jr., J. (2012). Guaranteeing motion safety for robots. *Autonomous Robots*, *32*(3), 173–175.

Dahlem Canter for Machine Learning & Robotics, (2017). Freie Universität Berlin, AutoModelCarWiki. Retrieved from https://github.com/AutoModelCar/AutoModelCarWiki/wiki/Hardware

Hopcroft, J., Motwani, R., & Ullman, J. (2000). *Introduction to automata theory, languages, and computation*. Boston, MA: Addison-Wesley Longman.

Hutchinson, S., Hager, G., & Corke, P. (1996, October). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, *12*, 651–670.

Latombe, J.-C. (1991). *Robot motion planning*. Norwell, MA: Kluwer Academic.

Laumond, J.-P. (1998). *Robot motion planning and control*. Berlin: Springer.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge: Cambridge University Press.

López-Nicolás, G., Gans, N., Bhattacharya, S., Sag"ués, C., Guerrero, J., & Hutchinson, S. (2011). An optimal homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *40*, 1115–1127.

Martinez, E., Laguna, G., Murrieta-Cid, R., Becerra, H. M., Lopez-Padilla, R., & LaValle, S. M. (2018). A motion strategy for exploration driven by an automaton activating feedback-based controllers, *Submitted to Autonomous Robots*, in second review.

Masutani, Y., Mikawa, M., Maru, N., & Miyazaki, F. (1994). *Visual servoing for non-holonomic mobile robots*. Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems, Munich, Germany.

Minguez, J., Lamiraux, F., & Laumond, J. P. (2008). Motion planning and obstacle avoidance. In Siciliano B and Khatib O (eds.), *Springer handbook of robotics* (pp. 827–852). Berlin: Springer.

Nunes, U., Laugier, C., & Trivedi, M. (2009, September). Introducing perception, planning, and navigation for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *10*(3), 375–379.

Ohya, A., Kosaka, A., & Kak, A. (1998, December). Vision-based navigation by a mobile robot with obstacle avoidance using a single-camera vision and ultrasonic sensing. *IEEE Transactions on Robotics and Automation*, *14*(6), 969–978.

Open, C. V. *Geometric image transformations*. Retrieved from https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html

Schwarting, W., Alonso-Mora, J., Paull, L., Karaman, S., & Rus, D. (2017). *Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention*. Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, Singapore.

Swain Oropeza, R., Devy, M., & Hutchinson, S. (2001). Sensor-based navigation in cluttered environments. *Proceedings of the IEEE/RSJInternational Conference on Intelligent Robots and Systems*, *1*, 1662–1669.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., . . . ,Mahoney, P. (2006, September). Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, *23*(9), 661–692.

Udacity, Inc., (2018). *Advanced lane finding*. Retrieved from https://github.com/ndrplz/self-driving-car/tree/master/project_4_advanced_lane_finding