

Vision-based Navigation System for Autonomous Urban Transport Vehicles in Outdoor Environments

M. A. Sotelo, F. J. Rodríguez

Department of Electronics
University of Alcalá
Alcalá de Henares, 28806, Madrid, Spain

L. Magdalena

Department of Mathematics
Technical University of Madrid
Madrid, 28040, Spain

Abstract

This paper describes a vision-based system for autonomous urban transport missions in outdoor environments. Vision-based specialised tasks are implemented for particular functionalities such as lane tracking, and navigation along intersections. High level knowledge about goals and intentions is extracted from an a priori map so as to schedule the global mission and to direct the behaviours of the low-level perception and actuation modules.

1. Introduction

We undertake the problem of intelligent unmanned mission execution for transport applications in unmarked urban like scenarios (university campus, industrial areas, etc) basing on DGPS and vision. In an attempt to adapt the algorithms to the already existing infrastructures, no limitation on the kind of road is imposed. According to this, two main challenges arise: lane tracking on non-structured roads (roads with no lane markers), and sharp turn manoeuvres in intersections (very usual in urban areas).

Previous works on this topic mainly focus on Intelligent Highway Systems. Nevertheless, many of the algorithms developed for structured roads are in part applicable to the detection and tracking of non-structured ones. A survey on the most remarkable works in this area primarily leads us to [14], [4], [10] and [1] where lane markers are successfully detected in the image plane for road position determination. An alternative approach considered in the NAVLAB project in the Carnegie Mellon University combines vision and learning techniques

(neurally inspired) to compute the characteristics that properly describe the path along the road [9]. Also in the Carnegie Mellon University the so-called "SAUSAGES" [13] control architecture was developed and tested on unmanned ground vehicles both at Carnegie Mellon and at Lockheed Martin. More recently, the ability to recognise intersections has been investigated in [8] in order to successfully navigate on unmarked road networks. Intersections were detected and tracked utilising an active pan-tilt head (TACC) to direct the focus of attention. Some promising results on autonomous turn-off manoeuvres conducted on unmarked campus roads were achieved and discussed in that work by using a four cameras arrangement.

2. Control Architecture

The system is intended to receive missions specifications. All modules on board the robot make use of the same description of the environment for several purposes dealing with mission specification and robot navigation. The environment model has been designed to ease the implementation of the mission execution system. The model is a topological and geometrical representation of the environment where no landmarks, neither natural nor artificial, have been used. Instead, global coordinates are exploited for geometrical recognition of relevant items such as intersections and stop stations. An environment is a topological graph of routes (arcs in the topological map), intersections, and stations (nodes in the topological map) obtained from the geometrical representation of the real working scenario, as depicted in figure 1. The routes are composed of lanes that have an exclusive direction.

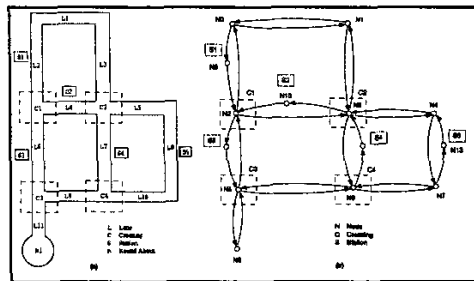


Figure 1. Environment representation. a) Geometrical model. b) Topological model.

Basically, the robot has to plan the shortest route to the destination station, making use of the topological environment model, and divide the execution of the resulting plan into several appropriate vision-based specialised tasks. As an example, a mission description can be specified as follows.

- T1: Track the lane until intersection C2
- T2: Turn left in intersection C2
- T3: Track the lane until destination station 3

The Robot Control Architecture is divided into three main layers (apart from sensors and actuators) as depicted in figure 2. A general description of each layer is given below.

Planning Layer: planning capability is issued by means of a global planner that makes use of a priori geometrical and topological information contained in the environment map. For planning purposes the location of relevant environmental features (such as intersections and stop stations) is needed. According to the topological representation of the environment we can treat the problem as one of traversing a mathematical graph structure, in which the graph edges represent tracks (or lanes) and the graph nodes represent the joins (intersections or stop stations) between tracks. For such a simple environment model the classical Dijkstra's algorithm [3] provides an efficient and fast solution to find the shortest route between the start and destination nodes in the topological graph. Along with the path, a velocity profile is computed issuing reference values for lanes and intersections traversing.

Coordination Layer: it basically includes the tasks manager and plays an essential role in the robot

control architecture. It manages and supervises the execution of all vision-based tasks (in a tasks scheduler manner), switching among them when appropriate while being subject to strong real time constraints. Each vision task has a termination condition that must be evaluated by the tasks manager in order to stop the current task and start the next one so as to guarantee correct plan execution.

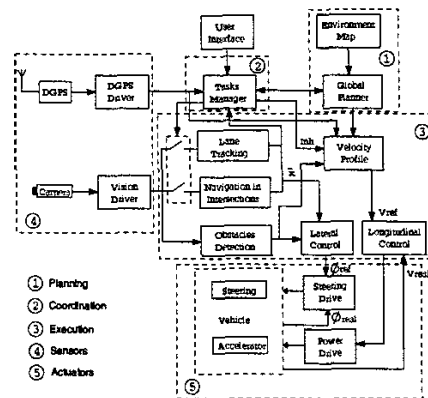


Figure 2. Robot Control Architecture.

Execution Layer: this level includes the following vision-based tasks: lane tracking, navigation on intersections, and vehicle detection. Their description is given in section 3. For simplicity, both the lateral and longitudinal controllers have also been included in this layer, although their detailed description has been omitted in this paper.

3. Vision-based tasks

As described above, the system exploits the well-known efficiency of particular processes in specialised tasks by switching between *Lane Tracking* and *Intersection Navigation* according to the plan, the environment model, and DGPS information.

3.1 Lane Tracking

The mission of this task is to provide correct lane tracking between two consecutive intersections.

Previous research groups [7] have widely demonstrated that the reconstruction of road geometry can be simplified by assumptions on its shape. Thus, a polynomial representation is used assuming that the road edges can be modelled as parabolas in the image plane. This model has been successfully proven by experience in previous works [12],[11]. Similarly, the assumption of smoothly varying lane width allows the enhancement of the search criterion, limiting the search of features to almost parallel edges. On the other hand, due to both physical and continuity constraints, the processing of the whole image is replaced by the analysis of a specific region of interest in which the relevant features are more likely to be found. All these well-known assumptions enhance and speed-up the road estimation processing [2].

According to these previous considerations the incoming image is on hardware re-scaled (the frame grabber itself performs this feature in real time), building a low resolution image of what we call the Area of Interest (AOI), comprising the nearest 20 m ahead of the vehicle. The AOI is segmented basing on colour properties and shape restrictions. The proposed segmentation relies on the HSI (Hue, Saturation, Intensity) colour space [6] because of its close relation to human perception of colours. The scheme performs in two steps:

1. Pixels are classified as chromatic or achromatic as a function of their HSI colour values according to equation 1.

$$I > 90 \text{ or } I < 10 \text{ or } S < 10 \quad (1)$$

where the saturation S and the intensity I values are normalised from 0 to 100.

2. Pixels are classified into road and non-road (including obstacles). Chromatic pixels are segmented using their HSI components: each pixel in the low resolution image is compared to a set of pattern pixels obtained in the first image in a non supervised manner. The distance measure used for comparing pixel colours is a cylindrical metric. It computes the distance between the projections of the pixel points on a

chromatic plane, as well as in the intensity axis, as defined in equations 2 and 3.

$$d_i = |I_s - I_i| \quad (2)$$

and

$$d_{ch} = \sqrt{(S_s)^2 + (S_i)^2 + 2S_s S_i \cos \Theta} \quad (3)$$

where

$$\Theta = \begin{cases} |H_s - H_i| & \text{if } |H_s - H_i| < \pi \\ 2\pi - |H_s - H_i| & \text{if } |H_s - H_i| > \pi \end{cases} \quad (4)$$

Subscript i stands for the pixel under consideration, while subscript s represents the pattern value. A pixel is associated to the road region if the value of metrics d_{ch} and d_i are respectively lower than thresholds T_{ch} and T_i . To account for road shape restrictions, threshold T_{ch} is affected by an exponentially decay factor yielding the new threshold value Γ that depends on the distance from the current pixel to the previously estimated road model, denoted by d as defined in equation 5.

$$\Gamma = e^{\frac{-Kd}{\hat{W}}} \cdot T_{ch} \quad (5)$$

where \hat{W} stands for the estimated width of the road and K is an empirically set parameter. This makes regions closest to the previous model be more likely to be segmented as road. For achromatic pixels, intensity is the only justified colour attribute that can be used when comparing pixels. A simple linear distance is applied in this case, so that the pixel is assigned to the road region if the difference is lower than a threshold value T_{ach} , similarly affected by an exponential factor, as equation 6 shows.

$$|I_s - I_i| < e^{\frac{-Kd}{\hat{W}}} \cdot T_{ach} \quad (6)$$

Once the segmentation is accomplished, a time-spatial filter removes non-consistent objects in the low resolution image, both in space and time (sporadic noise). After that, the maximum horizontal clearance (absence of non-road sections) is determined for each line in the AOI. The measured points are fed into a Least Squares Filter with Exponential Decay [12] that computes the road edges in the image plane as well as the central trajectory of the road using a second order

(parabolic) polynomial. A complete example is depicted in figure 3.

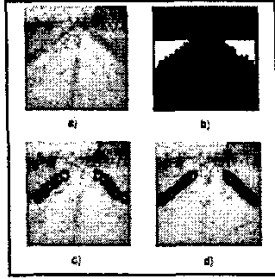


Figure 3. a) Original image. b) Road segmentation. c) Edges measure. d) Edges estimation.

3.2 Intersection Navigation

Intersections are conflicting and dangerous areas of the environment, in particular in urban areas where sharp turns must be carried out. To traverse an intersection the system fuses knowledge on the geometrical map, the plan under execution, and the incoming image, using in essence the same processing techniques described in the previous section to segment the road. The vehicle can perform three possible operations at an intersection: left turn, right turn, or go ahead

Left and Right turns: left and right turns require similar algorithmic treatment. One major difference with respect to lane tracking is that the road shape does not properly fit any polynomial model, in particular at the beginning of the curve (consider typical X or T-shaped intersections). That makes no sense to rely on it to estimate and update the road edges and width. This yields to consider two stages during the traversing of an intersection: in a first stage a slow turn (to the right or left depending on the plan) is issued until the vehicle starts to get properly positioned on the next street. The road shape doesn't fit the parabolic model during this stage. In a second stage the turn is completed at low speed by resuming lane tracking as soon as the road model fits the real road edges. Vehicle's localisation is required along the curve trajectory so as to decide when to resume lane tracking. To provide an estimation of the vehicle's position during the turn,

we model the arc described by the vehicle as a random variable ξ . Its probability density is maintained at any point in time in the framework of Markov localisation (a passive probabilistic approach to localization), successfully used for active localisation of indoor robots [5].

The key idea of Markov Localization is to compute a probability distribution (belief) over all possible locations (ξ) in the environment (in this case ξ can range from 0° , at the beginning of the curve, to 90° , for T-shaped intersections). $Bel(L_t=l)$ denotes the vehicle's belief of being at location l at time t . Here, l is a location in ξ space, being ξ the arc described by the vehicle along the curve. $Bel(L_0)$ reflects the initial state of knowledge: if the vehicle knows its starting position, $Bel(L_0)$ is centred on the correct location; if the vehicle does not know its initial location, $Bel(L_0)$ is uniformly distributed to reflect the global uncertainty of the vehicle. In our experiments, $Bel(L_0)$ is initially set to 0° taking advantage of the fact that the vehicle is starting the turn. The belief Bel is updated as follows.

The vehicle moves. Vehicle motion is modelled by the conditional probability $p_a(l|l')$. $p_a(l|l')$ denotes the probability that motion action a , when executed at l' , carries the vehicle to l . $p_a(l|l')$ is used to update the belief upon vehicle motion, where $\hat{Bel}(L_t=l)$ denotes the resulting belief at time t :

$$\hat{Bel}(L_t=l) = \sum_{l'} p_a(l|l') Bel(L_{t-1}=l') \quad (7)$$

In this work, $p_a(l|l')$ is computed accounting for the kinematic and dynamic constraints of the vehicle.

The vehicle measures. A sensor measure is denoted by s , so that $p(s|l)$ is the likelihood of measuring s at location l . Probability $p(s|l)$ specifies the probability of observations at the different locations during the turn. When measuring s , the belief is updated as follows:

$$Bel(L_t=l) = \frac{p(s|l) \hat{Bel}(L_t=l)}{p(s)} \quad (8)$$

where $p(s)$ is a normaliser to ensure that the belief Bel sums up to 1 over all l . Thus, the vehicle maintains a belief distribution $Bel(L)$ which is

updated upon vehicle motion, and upon sensor data measure. Such probabilistic representation allows to handle ambiguities and to represent degree-of-belief. Our implementation is based on a fine-grained geometric variant of Markov localisation where the angular resolution is 0.1 degree. The likelihood $p(s/l)$ is directly obtained from an approximate model of the environment. On the other hand, s is a vision-based measure that stands for the correlation between the current segmentation and several a priori road models obtained according to the parabolic-shaped constraint and the estimated road width. In other words, the higher the value of s , the higher the probability that the segmented road fits a parabolic model.

$$s = \max_i^N R_i, \quad (9)$$

where R_i stands for the correlation between the current segmentation and road model i , and N is the number of road models, empirically set to 5. This way vehicle localisation during the turn is enhanced by vision measures in a probabilistic manner. In case the degree-of-belief that the vehicle is located in the next street (and so the turn is about to complete) is high enough, normal lane tracking is resumed and vehicle's speed is increased according to its velocity profile. To illustrate the process figure 4 shows an example in which the vehicle traverses an intersection.

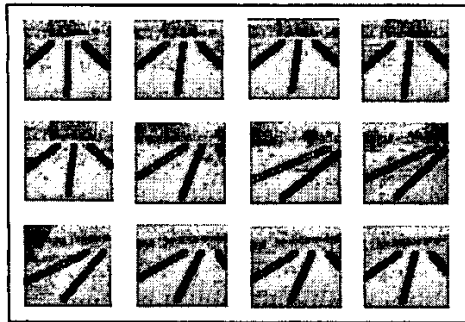


Figure 4. Intersection traversing example.

Go ahead: in many intersections the mission of the system is simply to traverse it and go ahead. This task requires different treatment than turning left or

right. In this case the incoming image is processed using the same algorithm described for lane tracking.

3.3 Vehicle Detection

In the first stage of this work we propose a monocular colour vision system for obstacles detection. According to the excellent discussion presented in [2] about vision-based obstacles detection for intelligent vehicles, using one single image leads to some limitations on the kind of obstacle that can be detected but provides a simple and fast method. We intend to detect other vehicles moving in the same or opposite lane. Using the road shape (given by the polynomial model) and an estimation of the road width (basing on the previous segmentation) the exact area of the image where the obstacles are expected to appear is determined. Thus, vehicles can be characterised by symmetry and edges features, within the estimated road, as far as usual vehicles have quite a distinguishable artificial shape and size that produces remarkable vertical edges in the filtered image. According to this, figure 5 depicts an example of vehicle detection along the run.

4. Implementation and results

All modules were developed in C under the Real Time Linux Operating System, running on a single PC (processing up to 15 frames/s). The system has been successfully tested on the electric vehicle (Citroën Berlingo commercial prototype) illustrated in figure 6, on a private test circuit in the *Instituto de Automática* of the *Consejo Superior de Investigaciones Científicas* (CSIC), where many successful autonomous missions have already been carried out on a network of non-structured roads. As an example, figure 7 shows the trajectory described by the vehicle during an autonomous mission. For further results, you can anonymously retrieve some compressed-AVI video files from <ftp://venus.depeca.alcala.es/pub/vision> exhibiting autonomous mission execution and vehicle detection capacities.

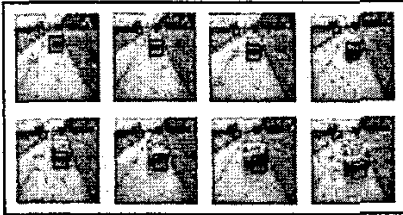


Figure 5. Vehicle detection example.

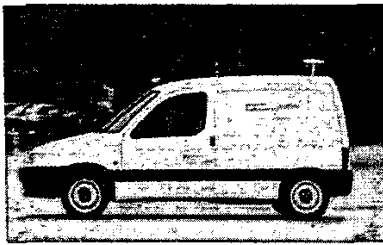


Figure 6. Commercial prototype used for testbeds.

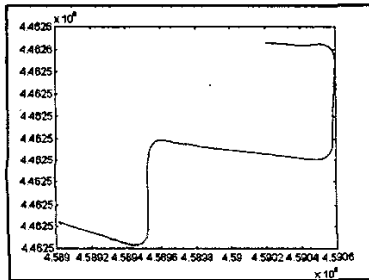


Figure 7. Vehicle trajectory during autonomous navigation from S5 to S1.

5. Future work

In the short term we intend to recognise intersections using visual information as well as to complement the vehicle detection system using radar sensors. Another important goal is to carry out overtaking manoeuvres using several autonomous vehicles.

Acknowledgments

The authors want to thank the generous support of the *Instituto de Automática* of the C.S.I.C for its crucial contribution to this work.

References

- [1] M. Bertozzi and A. Broggi. A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing* 7 (1) 62-81. 1998.
- [2] M. Bertozzi, A. Broggi, A. Fascioli. Vision-based intelligent vehicles: state of the art and perspectives. *Robotics and Autonomous systems* 32, 1-16. 2000.
- [3] S. Cameron and P. Probert. *Advanced Guided Vehicles. Aspects of the Oxford AGV Project.* World Scientific Publishing Co. Pte. Ltd, 1994.
- [4] E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrant, M. Mauer, F. Thomanek and J. Shiellhen. The seeing passenger car 'VaMoRs-P'. In *Proc. Of Int. Symp. On Intelligent Vehicles, Paris.* 1994.
- [5] W. B. Dieter and S. Thrun. Active markov localization for mobile robots. Preprint submitted to Elsevier Preprint. 1998.
- [6] N. Ikonomakis, K. N. Plataniotis and Venetsanopoulos. Color Image Segmentation for Multimedia Applications. *Journal of Intelligent and Robotic Systems.* 2000.
- [7] M. Lützel and E. D. Dickmanns. Road recognition with MarVEye, in: *Proceedings of the IEEE Intelligent Vehicles Symposium '98, Stuttgart, Germany, October 1998, pp. 341-346.*
- [8] M. Lützel and E. D. Dickmanns. EMS-Vision: Recognition of Intersections on Unmarked Road Networks. *International Symposium on Intelligent Vehicles (IV'2000).* Dearborn, (MI). October 2000.
- [9] D. A. Pomerleau. *Neural Network Perception for Mobile Robot Guidance.* Kluwer Academic Publishers. Boston. 1993.
- [10] D. A. Pomerleau and T. M. Jochem. Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert* 11 (2) 1996.
- [11] F. J. Rodríguez, M. Mazo, M. A. Sotelo. Automation of an industrial fork lift truck, guided by artificial vision in open environments. *Autonomous Robots* 5, 215-231. Kluwer Academic Publishers. 1998.
- [12] H. Scheneiderman and Marilyn Nashman. A discriminating feature tracker for vision-based autonomous driving. *IEEE Transactions on Robotics and Automation, Vol 10. NO 6,* 1994.
- [13] A. Stentz, M. Hebert and Chuck Thorpe. *Intelligent Unmanned Ground Vehicles. Autonomous Navigation Research at Carnegie Mellon.* Kluwer Academic Publishers. 1997.
- [14] C. Thorpe. *Vision and Navigation. The Carnegie Mellon Navlab.* Kluwer Academic Publishers. 1990.