VIRTUOUS: Vision-Based Road Transportation for Unmanned Operation on Urban-Like Scenarios

Miguel Angel Sotelo, *Member, IEEE*, Francisco Javier Rodriguez, *Member, IEEE*, and Luis Magdalena, *Member, IEEE*

***Abstract*—This work presents an intelligent transportation system (ITS) that was implemented on an autonomous vehicle designed to perform global navigation missions on a network of unmarked roads. This is the first step toward the complete implementation of ITS in urban environments, which is the long-term goal of this work. Using a global positioning system, global navigation is achieved by means of a global planner and a task manager that recurrently coordinate the execution of vision-based perception tasks for the road tracking of nonstructured roads and the navigation of intersections. In addition, a vision-based vehicle-detection task has been developed, which endows the global navigation system with a reactive capacity. The complete system has been tested on the BABIECA prototype vehicle, which was autonomously driven for hundreds of kilometers around a private circuit, designed to emulate an urban quarter, at speeds of up to 50 km/h, successfully carrying out different navigation missions. During the tests, the vehicle drove itself across crossroads and performed the appropriate turning maneuvers at intersections. It also demonstrated its robustness with regard to shadows, road texture, weather conditions, and changing illumination.**

***Index Terms*—Intelligent road transportation system, intersection navigation, unmarked roads, vision and global positioning system (GPS).**

I. Introduction

THE main issue addressed in this work is vision-based and differential global positioning system (DGPS) aided intelligent transportation systems (ITS) for the execution of autonomous missions on a network of unstructured roads that are designed to emulate an urban scenario.

A. ITS on Highways and Extra Urban Roads

The techniques deployed for road tracking on unmarked roads are, in many ways, similar to those developed for road tracking on highways and structured roads, as they face common problems. Nonetheless, most of the research groups currently working on ITS focus their efforts on autonomously navigating vehicles on structured roads; that is, marked roads. This reduces the navigation problem to the localization of lane markers painted on the road surface. This is the case with some well-known prestigious systems such as the rapid adapting lateral position handler (RALPH) [33], developed on the Navlab vehicle at the Robotics Institute, Canergie Mellon University, Pittsburgh, PA, and the impressive unmanned vehicles developed during the last decade by research groups at the Universitat der Bundeswehr Munchen (UMB), Munich, Germany, [17], [18], [28] and Daimler-Benz, Munich, Germany, [19] or the generic obstacle and lane detection (GOLD) system [3], [5], [6] implemented on the ARGO autonomous vehicle at the University of Parma, Parma, Italy. All of these systems have more than proved their validity in extensive tests carried out over thousands of kilometers of autonomous driving on structured highways and extra urban roads. Other interesting works on this topic, including offroad navigation, can be found in [10], [11], [15], [20], [25], [27], [31], [34], [37], [42], [44], and [45]. However, not that many research groups have tackled the problem of autonomous vision-based navigation on completely unstructured (unmarked) roads. Among the few that have are the supervised clustering applied to road following (SCARF) and unsupervised clustering applied to road following (UNSCARF) systems [46] and autonomous land vehicle in a neural net (ALVINN) [32]. Since the early 1980s, the group at the University of Bundeswehr, Munich, Germany, headed by Dickmanns, have also produced a remarkable amount of work on this topic. Early results for the autonomous guidance of vehicles on either marked or unmarked roads can be seen in [16] and [17], where nine road and vehicle parameters were recursively estimated following the four-dimensional (4-D) approach on three-dimensional (3-D) scenes. Nevertheless, despite some sound results obtained in this field, road-based vision on unmarked roads can still be regarded as an open problem at present.

B. ITS on Urban-Like Environments

Great interest has recently arisen in the design and development of intelligent systems for assisted driving, not only on highways but also in urban environments. In response to this, the urban traffic assistant (UTA) project [19], developed by the Daimler-Benz group, undertook the design of an intelligent stop-and-go system for inner-city traffic that uses stereo vision and demonstrates an ability to recognize traffic signs, traffic lights, walking pedestrians, zebra crossings, and stop lines. Other research groups have focused on partial problems, using vision-based pedestrian detection [48], [8], obstacle detection [7], [24], [30], or intersection detection [26], [38] to provide warning signals as aids to the human driver. A more ambitious project, aimed at recognizing intersections and autonomously navigating a vehicle across them, has been carried out at

Manuscript received August 8, 2002; revised October 27, 2003 and January 15, 2004. This work was supported by the Department of Electronics, University of Alcalá, Alcalá de Henares, Madrid, and the Industrial Automation Institute, Consejo Superior de Investigaciones Cientificas (CSIC). The Associate Editor for this paper was A. Broggi.

M. A. Sotelo and F. J. Rodriguez are with the Department of Electronics, University of Alcalá, Alcalá de Henares, Madrid 28872, Spain (e-mail: michael@depeca.uah.es; fjrs@depeca.uah.es).

L. Magdalena is with the Department of Mathematics, Technical University, Madrid 28040, Spain (e-mail: llayos@mat.upm.es).

Digital Object Identifier 10.1109/TITS.2004.828175

Fig. 1. Babieca autonomous vehicle.

Carnegie Mellon University. The first objective was achieved by means of a previously configured neural network, but the autonomous navigation of the intersection was only partially solved [22]. A similar system can be found in [29], where a truly autonomous system for intelligent navigation over a network of unmarked roads and intersections has been designed and implemented. The vehicle is equipped with the so-called expectation-based multifocal saccadic vision (EMS-Vision) system and can be considered to be the first completely autonomous vehicle capable of successfully performing global missions over a network of unmarked roads. The work developed by the Department of Electronics, University of Alcalá (UAH), Alcalá de Henares, Madrid, in the field of ITS began in 1993 with the design of a vision-based system for outdoor environments [37]. The system was implemented on an industrial forklift truck that was autonomously operated around the university campus. Close cooperation between the Department of Electronics and the Industrial Automation Institute (IAI), Consejo Superior de Investigaciones Cientificas (CSIC) since 1998 has led to the development of a vision- and DGPS-based ITS [43], [14] for the autonomous execution of global missions over a network of unstructured roads and intersections. This is the real innovation of the whole approach presented in this paper. The complete navigation system was implemented on BABIECA, the commercial electric prototype Citroen Berlingo depicted in Fig. 1. This vehicle is equipped with a color camera, a DGPS receiver, a computer, and the necessary electronic equipment to automatically control the steering wheel, as well as the brake and accelerator pedals. Thus, complete control of both lateral and longitudinal movement is automatically maintained during navigation. Real tests were carried out at the IAI on a private circuit designed to emulate an urban quarter with streets, intersections, and roundabouts. Additionally, a live demonstration exhibiting the system's autonomous driving capabilities was made during the IEEE Conference on Intelligent Vehicles 2002, on a private circuit in Satory, Versailles, France.

The work described in this paper is organized into the following sections. Section II presents the complete control architecture for the global navigation system. In Sections III and IV, the vision-based algorithms for lane tracking and intersection navigation, respectively, are described. Section V provides some global results and, finally, in Section VI the whole system and future projects are discussed and concluding remarks are made.

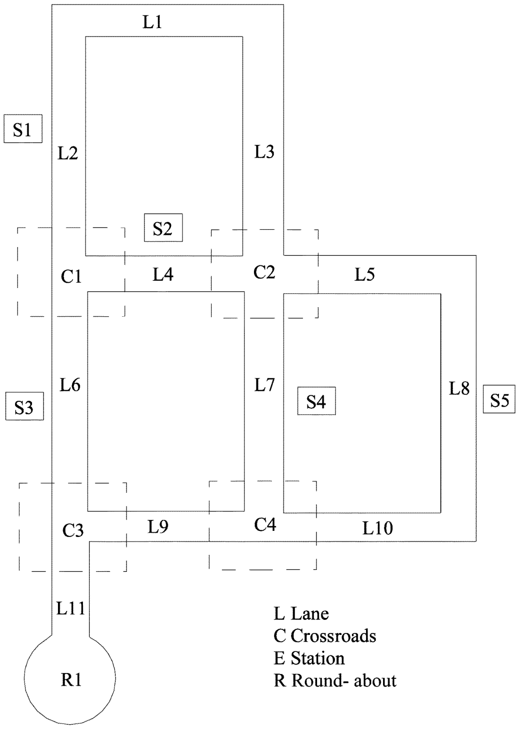


Fig. 2. Geometrical circuit representation.

II. Control Architecture

Efficient control architecture is needed to properly manage the information provided by the vehicle's sensors, (the color camera and DGPS receiver) and the flow of data generated during navigation. The design of the control architecture involves the development of a global system for task execution and monitoring to integrate the perception capabilities of the vehicle. A global planner is also needed to direct and focus the behavior of the perception and actuation modules on an environment model.

A. Environment Model

A geometrical and topological description is provided for the actual environment in which the vehicle operates. The aim of developing an environment model is to facilitate path planning, as can be seen in Fig. 2, where a geometrical map of the test circuit is depicted. The operating environment is designed to resemble an urban quarter, including streets, intersections, roundabouts, and stopping points. The geometrical map is a simplified version of the test circuit and does not take into account the real curvature at intersections.

The next step is to convert the geometrical map into a topologically directed graph, where both the intersections and stopping points are represented by nodes and the streets that link them by arcs (or edges) of different lengths, taking into consideration the direction of circulation, as in [1].

B. Control-Architecture Description

The control architecture has been divided into several classical layers, with the aim of planning and executing the optimal path between the current location and the destination stopping

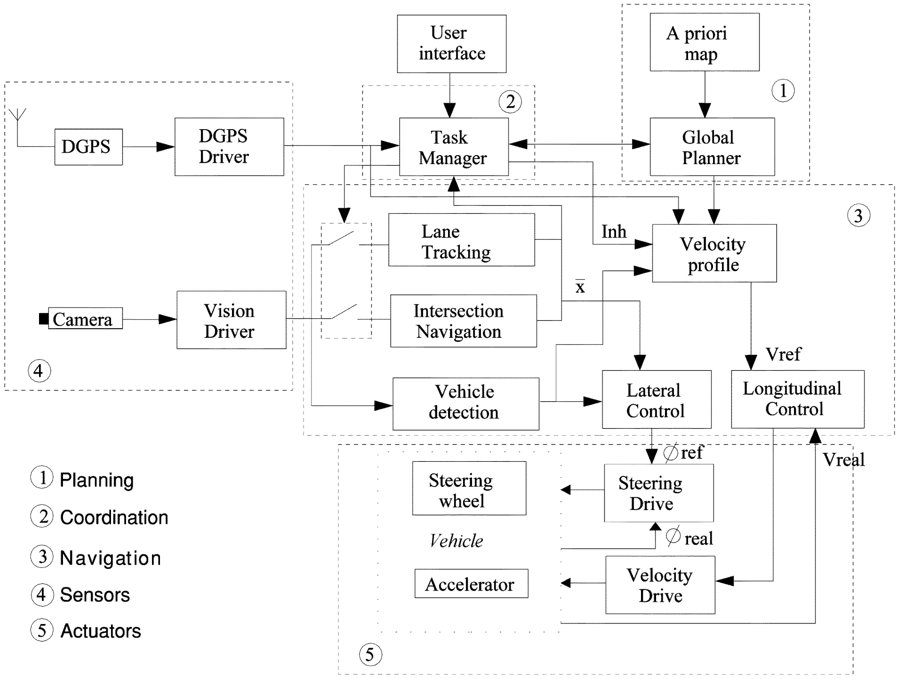


Fig. 3. Control architecture.

point specified by the user with the help of an *a priori* circuit map. Global navigation is achieved by properly concatenating local perception tasks that jointly solve vision-based navigation on streets and at intersections. The same idea was suggested and successfully deployed in [22] for crosscountry navigation. The proposed architecture is shown in Fig. 3. A basic description of the different layers of the control scheme is provided as follows.

- Planning layer: the global planner included in this layer computes the shortest path between the current location and the destination station, providing a recommended velocity profile for the global mission depending on whether the vehicle is navigating along a street or at an intersection.
- Coordination layer: the core of this layer is the task manager and it provides a link between planning and execution by enabling the system to manage tasks and replan its path in emergency situations or at explicit user request.
- Navigation layer includes vision-based tasks for lane tracking, intersection navigation, and vehicle detection.
- Low level is composed of the vehicle's onboard sensors, a color camera, and DGPS receiver, together with their respective synchronized software drivers and the actuator modules for the steering wheel and acceleration pedal.

C. Global Planner

Depending on the previously obtained topological model of the environment, the path-planning problem can be reduced to one of traversing a mathematical graph composed of arcs, or edges, and nodes, where the edges represent tracks (or streets) and the nodes represent the intersections. To find the shortest route on this graph, the popular Dijkstra algorithm [9] has been chosen. Although the shortest route may not always be the best option, we have decided to adopt this simple criterion for demonstrative purposes. Other criteria or objectives can be easily accommodated by simply modifying the analytical expression used to evaluate the different solutions found by the algorithm during execution. Likewise, the global planner provides an appropriate velocity profile for the different sections of the route by taking into account the kinematic and dynamic constraints of the vehicle. Local navigation of each section of the final route involves the following specialized vision based tasks: lane tracking and intersection navigation. A global plan could look something like

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

III. Lane Tracking

The main aim of this vision-based task is to correctly track the lane of any kind of nonstructured road (roads without lane markers painted on them), while correctly detecting other vehicles. Efficient performance of this task is essential if the system is to autonomously navigate between two intersections in a reliable and accurate manner. Given the wide range of weather and illumination conditions in outdoor scenarios, vision-based lane tracking on unmarked roads can be regarded as a complex problem to tackle. Both the lane-tracking system and the vision-based vehicle-detection algorithm (developed in this work) have already been extensively documented in [43]. However, a summary of the lane-tracking system is provided in this paper.

A. Road Segmentation

As has been successfully tested in previous work [39], the lane-tracking task begins with the use of a second-order polynomial road model for both the edges and center of the road.

Road models are used because they ease the reconstruction of road geometry and permit the filtering of data computed during the feature searching process. Then, making use of the system's hardware capabilities, the original 480×512 incoming image acquired by the color camera is rescaled in real time to a low-resolution 60×64 image. Accordingly, all figures in this paper that show the result of a vision-based process are low-resolution images of 60×64 pixels. The aim of this process is to decrease computing time, although it is, of course, dependant on the real-time constraints implicit in the control of high-speed vehicles for road-tracking applications. As discussed in [4], due to the existence of physical and continuity constraints derived from vehicle motion and road design, it is more efficient to analyze a specific portion of the image, namely, the region of interest, rather than the whole image. Accordingly, a rectangular region covering the nearest 20 m ahead of the vehicle is proposed. This restriction is sufficient to ensure the early detection of other vehicles, particularly in urban or industrial areas where the maximum velocity usually is under 50 km/h.

The combined use of color and shape restrictions provides the essential information required to drive along nonstructured roads. Prior to the segmentation of the image, the accurate selection of the most suitable color space becomes an essential part of the process. Although the red, green, blue (RGB) color space has been extensively tested and used on nonstructured roads in previous road-tracking applications [46], [13], [37], it does have some well-documented disadvantages; it is nonintuitive and does not separate colors uniformly, which means that two relatively close colors may be widely separated in the RGB color space, the components of which are only slightly correlated. This means that humans cannot imagine a color from its RGB components. In some applications, the RGB color information is changed into a different color space where the luminance and chrominance components of the color are clearly separated from each other. It also has the advantage that the color-description model is reasonably similar to human perception of colors. Furthermore, in outdoor environments, changes in luminance can be substantial due to unpredictable and uncontrollable weather conditions, while changes in color or chrominance are not that relevant. This makes it highly recommended to use a color space where a clear separation between intensity (luminance) and color (chrominance) information can be established. The hue, saturation, and intensity (HSI) color space is a good example, as it allows colors to be described in terms that can be intuitively understood. Humans can easily recognize the basic color attributes, such as intensity (luminance or brightness), hue of color, and saturation [21], [23]. Hue represents the impression made by the predominant wavelength in the perceived color stimulus. Saturation corresponds to the relative purity of the color and, thus, nonsaturated colors are grayscale colors. Intensity is the amount of light in a color; maximum intensity is perceived as pure white and minimum intensity as pure black. Some of the most relevant advantages of using the HSI color space are that it is close to human perception of colors, having a advanced ability to discriminate between colors, especially hues. The difference between colors is directly quantified by using a distance measure. For these reasons, we propose using the color features of the HSI color space as the basis for performing the segmentation of nonstructured roads. The HSI color space segments the image by using the cylindrical distribution of its color features. From the analytical point of view, the difference between two color vectors in the HSI space can be established by computing the distance that separates them along the chromatic plane and the intensity axis. Before segmenting the image, pixels are divided into chromatic and achromatic, as proposed in [23]. Achromatic pixels are only segmented according to their intensity value. Obviously, nonachromatic pixels are automatically categorized as chromatic. The chromatic pixels are segmented using both their chromatic and intensity distances to a given road-pattern HSI color vector. The HSI color features of road pixels are confined to a cylindrical region around the HSI road pattern vector. The road-pattern HSI color vector is updated during every iteration, as described in Section III-D.

The quality of road segmentation can be strongly enhanced by adding spatial constraints based on the geometrical road model. In an intuitive approach, the probability of a pixel being segmented as road is high if it is located close to the central trajectory of the previous road model (as estimated during the last iteration of the algorithm). In order to incorporate spatial constraints to the segmentation of each pixel in the image, the dimension of the cylindrical surface used for segmentation, denoted by d, is modified according to the geometrical distance in the three-dimensional (3-D) space between the pixel under consideration and the previously estimated road model. The distance is computed from measurements calculated along the image plane using the camera calibration parameters under the flat terrain assumption, thus turning the segmentation stage into a position- and color-dependant process. This is based on the way in which humans perceive the road while driving: those areas of the scene that resemble the road, such as paths running parallel to it, but are in fact unlike our previous model of it, are disregarded. Accordingly, those pixels near the central trajectory of the previous road model are segmented using the proposed modification, so that low threshold values are achieved. For those pixels far away from the previous road model, the opposite is true and then the values are higher. From the analytical point of view, the proposal reflects an exponential variation of threshold values T_{chrom} and T_{int} (for chromatic distance d_{chrom} and intensity distance d_{int}, respectively) for each individual pixel i as a function of d, according to the expression in

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

where $\Psi_c(d)$ and $\Psi_I(d)$ represent the threshold values for the chromatic and luminance distances, respectively. For a pixel located at a distance of d from the previous model, $T_c(t-1)$ and $T_I(t-1)$ are the maximum threshold values estimated during the previous iteration. K is an empirically determined parameter devised to control the threshold value, particularly of those pixels located at the sides of the road. The value of K is calculated based on the empirical fact that the distribution of the color distance between road pixels and the road pattern has a standard deviation that is below 30% in most of the real cases. This leads us to intuitively select a color threshold based on a similarity greater than 70%. In practice, K is determined so that the threshold value is 70% of the maximum threshold for pixels located at a certain distance from the model $d_w = \hat{W}(t-1)/2$, where $\hat{W}(t-1)$ represents the estimated road width during the previous iteration of the algorithm. This empirical value has demonstrated its appropriateness in practical trials by helping to carry out a stable segmentation, yielding the numerical value for K depicted

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

A chromatic pixel i is classified as road if it simultaneously verifies that $d_{\text{chrom}} < \Psi_c(d)$ and $d_{\text{int}} < \Psi_I(d)$, while an achromatic pixel is segmented as road if the single condition $d_{\text{int}} < \Psi_I(d)$ is satisfied. On the other hand, the maximum threshold values $T_c(t)$ and $T_I(t)$ must be dynamically updated so the segmentation process can adapt to changing color and luminance conditions. To achieve this, the root-mean-squared values of the chromatic and luminance distances to the road pattern ($d_{\text{chrom,rms}}$ and $d_{\text{int,rms}}$) are computed for each pixel classified as road. The maximum threshold values for the next iteration ($T_c(t+1)$ and $T_I(t+1)$) are calculated as described in (3), dnding on $d_{\text{chrom,rms}}(t)$, $d_{\text{int,rms}}(t)$, and an exponential factor essential to guarantee the stability of the segmentation process. This provides the threshold values for the next iteration ($\Psi_c(d)_{|t+1}$ and $\Psi_I(d)_{|t+1}$), exactly equal to $d_{\text{chrom,rms}}(t)$ and $d_{\text{int,rms}}(t)$, respectively, for pixels located on the road edges ($d_{|t+1} = \hat{W}(t)/2$).

$$\begin{aligned} T_c(t+1) &= d_{\text{chrom,rms}}(t) \cdot e^{\frac{K}{2}} \\ T_I(t+1) &= d_{\text{int,rms}}(t) \cdot e^{\frac{K}{2}}. \end{aligned} \tag{3}$$

To enhance the quality of the segmentation process, the resulting binary image is reinforced by a morphological opening operation followed by the removal of small white blobs caused by segmentation noise.

B. Handling Shadows and Brightness

Shadows and bright spots on the road are admittedly one of the greatest difficulties for vision-based systems operating in outdoor environments [3]. The problem becomes especially dangerous at the hours of the day when the sun shines directly onto the image plane, causing loss of tracking, or when entering or exiting tunnels, which has the same effect. To tackle this problem of strong luminance changes, some authors propose improving the dynamic range of visual cameras [4] or enhancing the sensitivity of the cameras to the blue component in colors. Another approach looks only at the problem of shadows, attenuating their effects by using an appropriate software preprocessing technique to relay the physical properties of shaded road pixels. This means that the resulting segmentation can be enhanced against the effects of both shadow and brightness. The color features of pixels located within the limits of the road, but classified as nonroad after the segmentation process, are considered for brightness and shadow attenuation. At the beginning, the shaded pixels should simultaneously exhibit an intensity value lower than average for road pixels, while presenting a predominantly normalized blue component. The pixels that meet these conditions are assumed to belong to a shadow on the pavement and, consequently, are reclassified as road pixels. This technique allows the road segmentation to be enhanced in the presence of shadows and contributes enormously to improving the robustness of the color-adaptation process, particularly along stretches of road that are largely in shadow. Although these assumptions are quite simplistic, they have demonstrated in practice to be extremely useful in attenuating shadows and brightness, achieving higher quality segmentations. Even in other situations that produce similar effects on the image, such as potholes, the attenuation process would still remove them from the segmented image, leading to clear road segmentation where edge features can be easily extracted. On the other hand, alternative sensorial systems, such as stereovision, would be needed to accurately detect the presence of potholes and assess, by measuring their depth, for example, the danger they represent. To graphically illustrate the benefits derived from this operation, Fig. 4 shows an example of road segmentation in the presence of strong shadows. As can be appreciated, the road edges are neatly distinguished after the attenuation of the shadows.

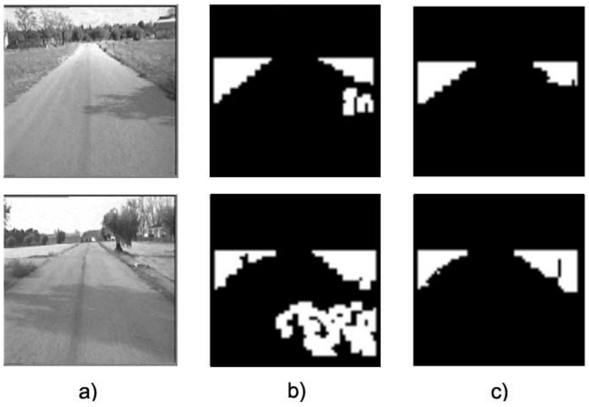


Fig. 4. Attenuation of shadows. (a) Original shaded images, (b) segmentation without attenuation of shadows, and (c) segmentation after attenuation of shadows.

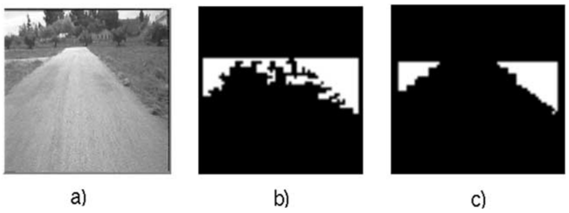


Fig. 5. Brightness attenuation. (a) Original image with brightness on the pavement, (b) segmentation without brightness attenuation, and (c) segmentation after brightness attenuation.

Analogously, a brightness attenuation technique has been devised. In this case, pixels initially classified as nonroad, but located within the road edges and exhibiting higher intensity values than the average road pixels, are assumed to correspond to brightness on the pavement caused by the sun and, consequently, are reclassified as road pixels. After applying the previous process, white blobs, caused by brightness, are removed from the segmentation, as depicted in Fig. 5. The improvement achieved by attenuating both brightness and shadows as described allows real images, in real and complex situations, to be

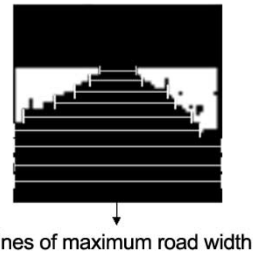


Fig. 6. Maximum road width for each line in the region of interest.

handled at an extraordinarily high performance level, making it an outstanding feature of this work.

C. Estimation of Road Edges and Width

As previously described, the central trajectory of the road and its edges are estimated using parabolic functions. These polynomial functions are needed to obtain the lateral and orientation errors of the vehicle with respect to the center of the road.

1) Estimation of the Central Trajectory of the Road: For the central trajectory of the road in Fig. 5(a), the current instant in time is estimated using the segmented low-resolution image and the previously estimated road trajectory. To enhance the road-estimation process, the temporal integration of the measures obtained at different instants of time and the amount of data used to carry out the estimation is analyzed. As initially proposed in [40], a weighted-recursive least-squares estimator with exponential decay is used for this purpose.

a) Data Measurement and Validation: The aim of this first stage is to extract a number of candidate points from the central trajectory of the road at time instant t. For each line k in the region of interest, the maximum road width is determined using the segmented image, as depicted in Fig. 6. The middle point of each maximal road width line k is considered to be a candidate and its coordinates $(y_{t,k}, x_{t,k})$ are validated if the road width of line k is greater than a previously computed threshold. In order to provide the algorithm with noise-rejection capacity, only those candidate pixels whose distance d from the previous estimation of the road center (denoted by $\hat{y}_c(t-1)$), in the 3-D scene under the threshold V are validated and associated with the current measurement of the central trajectory of the road. This allows for a validation area to be established around the previous road model. All measurements outside the validation area are regarded as invalid and are discarded.

b) Road-Model Update: Measures validated in the last stage constitute the starting point for updating the parabolic road model. As previously mentioned, to estimate the central trajectory of the road, a weighted-recursive least-squares estimator with exponential decay is proposed. Although it is a well-known and well-documented theory, we repeat the basic equations in this paper for completeness. Thus, for a number of N_t candidate edge points at time t with coordinates in the image plane given by (y_t, x_t), the estimation of the parabolic function coefficients (a, b, c) that describe the road model in the image plane is carried out in the three stages described as follows.

a) Update prediction

$$\hat{z}(t) = \phi^T(t) \cdot \theta(t-1). \tag{4}$$

b) Update state-covariance estimate

$$P(t) = \frac{1}{\lambda}[P(t-1) - [P(t-1)\phi(t) \cdot (\lambda I + \phi^T(t)P(t-1)\phi(t))^{-1} \cdot \phi^T(t)P^T(t-1)]]. \tag{5}$$

c) Update state estimate

$$\theta(t) = \theta(t-1) + G(t) \cdot [z(t) - \hat{z}(t)] \tag{6}$$

with

$$G(t) = P(t-1)\phi(t) \cdot (\lambda I + \phi^T(t)P(t-1)\phi(t))^{-1}$$

$$z(t) = \begin{bmatrix} y_{t,1} \\ y_{t,2} \\ \cdots \\ y_{t,N_t} \end{bmatrix} \quad \theta(t-1) = \begin{bmatrix} c \\ b \\ a \end{bmatrix}$$

$$\phi^T(t) = \begin{bmatrix} 1 & x_{t,1} & x_{t,1}^2 \\ 1 & x_{t,2} & x_{t,2}^2 \\ \cdots & \cdots & \cdots \\ 1 & x_{t,N_t} & x_{t,N_t}^2 \end{bmatrix}$$

where $\theta(t-1)$ represents the state estimation during the previous iteration of the algorithm, i.e., at time instant $t - \Delta T_v$ (ΔT_v is the sampling period of the road tracking algorithm). λ is a scalar value that can vary in the range $0 \leq \lambda \leq 1$ and $P(t-1)$ represents the state covariance during the previous time step. To achieve a proper tradeoff between robustness and transient response, λ has been experimentally set to 0.7, performing adequately in real tests.

2) Road-Edge Estimation: Road edges are estimated using the same filtering technique described in the previous section. Measures for the left and right road edges are validated and enhanced using three fundamental points: the estimation of the central trajectory of the road at current time t; the estimation of road width at time $t-1, \hat{W}(t-1)$; and the slowly varying road width assumption. Thus, a validation area is established for both the left and right measures. The location of the left and right validation areas are based on the central trajectory of the road estimated at time t and the estimated width at time $t-1$. The left validation area is situated on the left, $\hat{W}(t-1)/2$ m from the central trajectory of the road $\hat{y}_c(t)$, while the right validation area is obviously located on the right, at $\hat{W}(t-1)/2$ m from $\hat{y}_c(t)$. The left and right edges are estimated independently using the validated measures, obtained by the same weighted-recursive least-squares approach.

3) Road-Width Estimation: The road width is estimated using the previously mentioned slowly varying road-width assumption. An individual road width measure is obtained for each line in the region of interest, by computing the difference between the left and right edges. The average road width measure at time $t, W(t)$ is computed using the individual measures for each line, normalized by the number of valid measures in the region of interest. The slowly varying road-width assumption is incorporated using a recursive least-squares-based estimator, similar to those employed for the estimation of road edges. This allows for a smooth estimation of the road width.

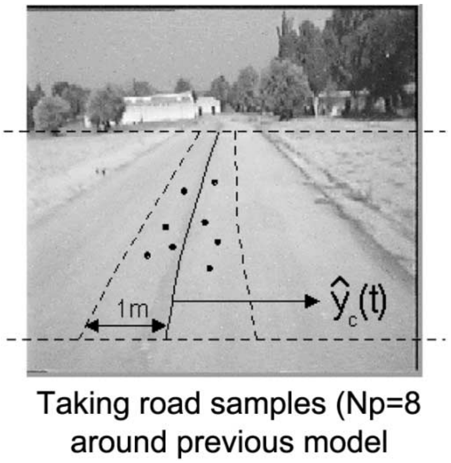


Fig. 7. Random selection of pixels for HSI road-pattern update.

D. Road-Color Features Update

After completing the road-edges and width-estimation processes, the HSI color features of the road pattern are updated to take into account changes in road appearance and illumination. Thus, the updated road-pattern HSI color vector will serve as the basis for the road-segmentation process in the next iteration of the algorithm, as described in Section III-A. Pixels close to the skeleton lines of the road intuitively present color features that closely resemble the road color pattern. Accordingly, as depicted in Fig. 7, a set of $N_p = 8$ pixels in a region of 1 m in the 3-D scene surrounding the central estimation of the road during the current iteration, denoted by $\hat{y}_c(t)$, is chosen at random. Obviously, the selected pixels are only validated if they have been segmented as road pixels during the current iteration.

As previously mentioned, the HSI road-pattern characteristics for the next iteration of the algorithm are then computed by correctly averaging the individual HSI characteristics of the pixels selected for the current iteration. The adaptation process described in this section is, in practice, crucial for ensuring a stable performance by the segmentation algorithm during changing illumination conditions and on color-varying asphalt.

E. Discussion of the Method

Performance of the lane-tracking system has been extensively analyzed on several nonstructured roads with extreme changes in both weather and lighting conditions. The road-tracking algorithm was initially evaluated on a private circuit of unmarked roads and on rural roads, obtaining correct results in both cases. Nonetheless, a more realistic testing ground for evaluating the system is a university campus, where typical urban conditions, such as zebra crossings, parked vehicles, etc., exist. Fig. 8 depicts two representative situations of urban driving. Fig. 8(a) shows the segmentation and road-edge estimation in the vicinity of a zebra crossing, while Fig. 8(b) illustrates the results obtained close to other vehicles. Correct segmentation and edge estimations are also achieved in both cases. The road-tracking scheme was also evaluated on roads without asphalt, yielding adequate results even in this kind of scenario. To verify the validity and generality of the segmentation and updating scheme, the road-tracking algorithm has been tested under different environmental and weather conditions, such as sunny, cloudy, rainy, and even foggy conditions. Some examples of segmentation and road-edge estimations obtained under these conditions are given in Figs. 9–11.

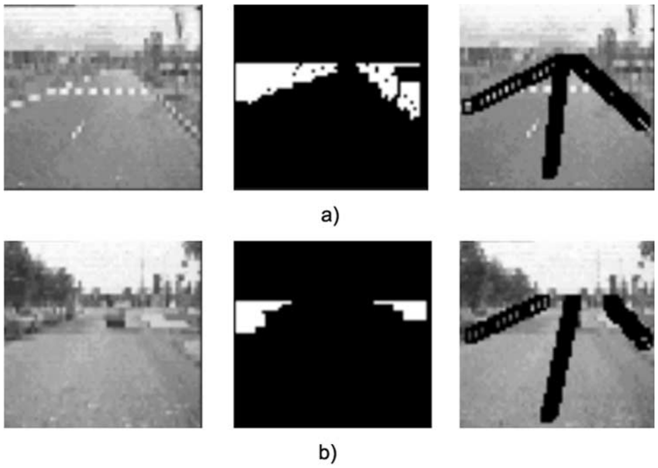


Fig. 8. Segmentation and road-edge estimation in urban areas in the presence of (a) a zebra crossing and (b) other vehicles.

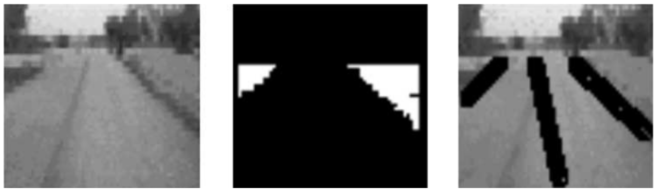


Fig. 9. Segmentation and edge estimation under cloudy conditions.

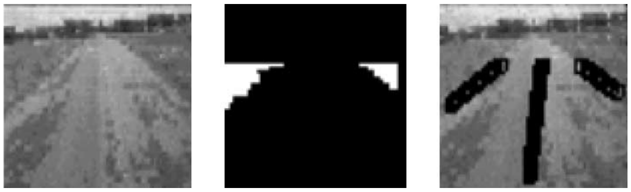


Fig. 10. Segmentation and road-edge estimation under post-rainy conditions.

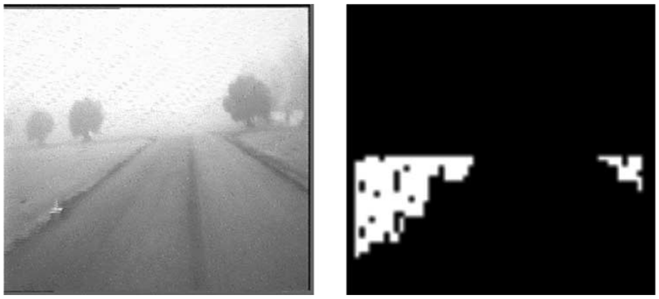


Fig. 11. Road segmentation under nonthick foggy conditions.

The previous section lets us state that the road-segmentation algorithm based on the HSI color space and under 2-D spatial constraints successfully provides an accurate and robust estimation of the edges and width of nonstructured roads, i.e., roads without lane markers. As demonstrated, the practical results also support the validity of the method under different environmental and weather conditions. There are, however, some limitations to the use of the road-tracking algorithm just after sunrise, just before sunset, and on very sunny days, primarily due to the direct incidence of sun rays onto the camera plane.

IV. Navigation at Intersections

Intersection navigation is completely vision based and accounts for any angular value between the intersection branches. The same system used for road tracking, a simple monocular color-vision system, is proposed for navigating at intersections. This is a major issue, as it will make cheap prototype navigation systems possible. Two basic maneuvers can be executed at an intersection: it can change its direction by turning left or right or it can go straight ahead and cross the intersection by maintaining its current direction. As the problem of crossing an intersection is basically the same as that of tracking the lane, even though some road edges are partially or completely occluded, the same algorithmic solution is used for this kind of maneuver. This is possible thanks to the ability of the lane-tracking algorithm, using the temporal recurrent least-squares filtering technique previously described, as it deals with occlusions of the road-edge features, even during several consecutive iterations. If, for example the left road edge disappears from the scene, the right road edge, together with the estimation of the road width, serves as the basis for reconstructing the road geometry. Turning right or left at an intersection is quite a different problem that needs to be addressed in a different manner.

A. Turning Maneuvers at Intersections

The scientific rationale for the navigation strategy proposed for turning maneuvers made at intersections is based on the way in which humans drive. So, let us consider the case of a human driver carrying out a turning maneuver at an intersection with strongly reduced visibility. The human driver would probably start the turning maneuver, left or right, at a very low velocity until complete visibility of the new road was attained, from which point increasingly higher velocities could be reliably achieved. Taking into consideration the limitations of the camera's field of vision, a similar approach could be used to make an autonomous vehicle perform a turning maneuver at an intersection. The vehicle should begin the maneuver (left or right, according to the plan) at a low velocity. Throughout the turn, the vehicle performs a simple circular open-loop trajectory described by its minimum radius of curvature, which is given by a maximum steering angle of 30°. The maneuver is carried out at a low speed until a large enough perspective of the new road is gained. From that point on, lane tracking resumes control of the vehicle and its velocity gradually increases. Due to the limited perspective of the road, particularly at the beginning of the turning maneuvers, the road edges can no longer be modeled as parabolic polynomials. In this case, there is no correspondence between the real road edges and a second-order function, particularly when the road edges are not even within the camera's field of vision. However, this should not cause an abrupt discontinuity in the road-model estimation when traversing an intersection. To solve this contradictory and compromising situation, we propose decoupling the segmentation process from the road-edge estimation. This leads to the use of a fixed road model that is not updated with the image segmentation results obtained during the turn. This invariable model is the valid reference used to plot the vehicle's turning angle, until a large enough perspective of the road is gained and road tracking can be resumed. This makes segmentation and road-edge estimation independent from each other during the turn at an intersection. The fixed road model is geometrically located in the image plane facing the direction, chosen in the global plan, of the turn to be performed at the next intersection. Fig. 12 shows the fixed models for left and right turns at an intersection, which cause the vehicle to turn left or right, respectively. These models are located in the region of the image where the road is expected to reappear when the turn has been completed, so no discontinuity in the vehicle-turning angle occurs. On the other hand, the road-width estimation is also kept constant throughout the entire turn, taking as constant the last estimated value before starting the turn $W(t_0)$.

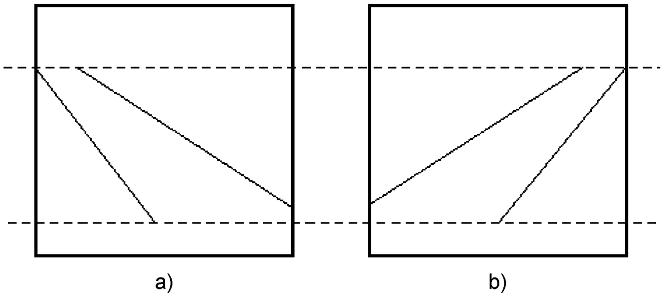


Fig. 12. Fixed road models for (a) left and (b) right turns at intersections.

Continuity in the road-edge and -width estimation is preserved due to the least-squares-based estimator that brings the road model from its initial position at the beginning of the turning maneuver (at $t = t_0$) to the corresponding fixed model in a soft and gradual manner. As previously mentioned, the end of the turning maneuver is basically determined by visual information. For this reason, the image segmentation is correlated with several *a priori* road models. The vehicle is expected to appropriately perceive the new road when the previous correlation is high enough, i.e., when the new road resembles some of the *a priori* road models utilized in the comparison. From that point onward, lane tracking resumes control of navigation.

1) Image Processing at Intersections: Basically, except for those processes concerned with feature adaptation as a function of the road model, image processing at intersections is similar to image processing for lane tracking. Indeed, the road model cannot be used for HSI feature updating during a turning maneuver at an intersection, as it remains fixed and so provides no relevant information. Accordingly, segmentation is performed solely on the basis of HSI color characteristics and no threshold modification is carried out based on the distance between the pixel under consideration and the estimated road model. The rest of the image-segmentation process remains unchanged. Finally, the HSI road-color pattern is updated based on all the pixels segmented as road during the current iteration, thus avoiding the need to rely on the road model, as it provides no reliable information during the turn. Experience shows that the image-segmentation and adaptation method presented in this section, al-

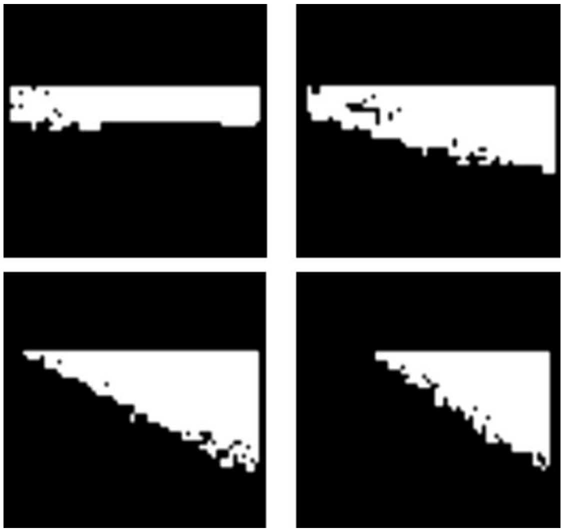


Fig. 13. Segmented images of a left turn at an intersection.

though slightly degraded, still remains stable during intersection navigation as long as there is enough road within the camera's field of vision. To demonstrate the performance of the segmentation process under these conditions, a sequence of four segmented images of a left turn at an intersection is depicted in Fig. 13.

B. Determining the End of the Turning Maneuver

As previously described, the end of the turning maneuver is determined using visual data; the maneuver finishes when enough of the new road comes into view. So, in a first intuitive approach, the end of the turning maneuver is determined by computing the correlation between the incoming segmentation and some *a priori* road-model templates. Unfortunately, experience shows us that this simple correlation measure cannot reliably determine the end of the turn. On the one hand, segmentations similar to the *a priori* road-model templates can occur even from the beginning of the turn, which could lead to a false detection. On the other hand, without a parabolic model to enhance the segmentation, segmentation noise increases drastically during the turn. For this reason, vehicle localization during the turn is reinforced using a Markov stochastic process, which from here on will be referred to as Markov localization process. The basic idea is to enhance vehicle-localization robustness while continuing to use visual information. For this purpose, the angular trajectory followed by the vehicle during the turn is modeled by a random variable denoted by ξ, as depicted in Fig. 14.

As is graphically shown by variable ξ in Fig. 14, the localization space is defined as all the possible angular positions of the vehicle. Its definition domain ranges from 0° at the beginning of the turn to 90°, or higher for really sharp turns, by the end of the turning maneuver. A probability density function (pdf) is calculated for all possible positions along the localization space. The function is updated during each iteration using the typical Markov assumptions and in so doing becomes a Markov stochastic process. The abcise $\xi_{\max}$, where the density function reaches its maximum, indicates the vehicle's most reliable angular position during the turn. Let $\mathrm{Bel}(\xi_t = \xi)$ denote the probability of being at location ξ at time t, where ξ is a location within the localization space. $\mathrm{Bel}(\xi_0)$ reflects the initial state of knowledge. If the vehicle's position is accurately known, $\mathrm{Bel}(\xi_0)$ is centered on it. If not, $\mathrm{Bel}(\xi_0)$ is uniformly distributed to reflect the global uncertainty about the vehicle's location. In this work, $\mathrm{Bel}(\xi_0)$ is initially set to 0°, taking advantage of the fact that the vehicle is starting the turn. The distribution $\mathrm{Bel}(\xi)$ is updated whenever the vehicle moves or acquires a new image.

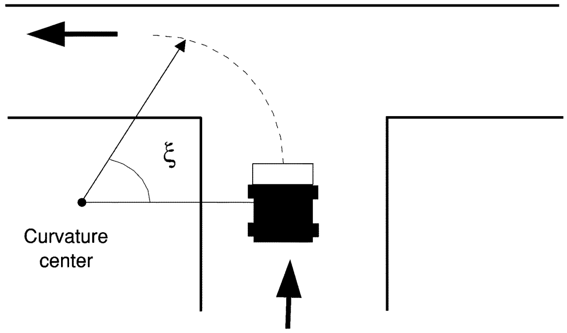


Fig. 14. Modeling of the vehicle's turning angle ξ at intersections.

1) $\mathrm{Bel}(\xi)$ *Updating During Vehicle Movement:* Vehicle motion is modeled by the conditional probability $p_a(\xi \,|\, \xi')$. $p_a(\xi \,|\, \xi')$ denotes the probability that motion action a, when executed at ξ', carries the vehicle to position ξ. $p_a(\xi \,|\, \xi')$ is then used to update the belief function about vehicle motion, where $\widehat{\mathrm{Bel}}(\xi_t = \xi)$ denotes the resulting belief at time t, as indicated in

$$\widehat{\mathrm{Bel}}(\xi_t = \xi) = \sum_{\xi'} p_a(\xi \,|\, \xi') \cdot \mathrm{Bel}(\xi_{t-1} = \xi'). \qquad (7)$$

Computation of $p_a(\xi \,|\, \xi')$ is carried out, taking into account the kinematic and dynamic constraints on the vehicle. This implies the use of the vehicle kinematic model (which is similar to the popular Ackermann model) and proprioceptive knowledge about the vehicle's current velocity v and steering angle ϕ. Let R denote the radius of curvature of the trajectory followed by the vehicle during the turn. The differential angular arc $\Delta\xi$, described by the vehicle between two consecutive iterations, can easily be obtained, as in (8), where the vehicle's linear velocity v is kept constant.

$$\Delta\xi = \frac{\Delta l}{R} = \frac{v \cdot \Delta t}{R} \qquad (8)$$

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

$$R = \frac{L}{\tan\phi} \qquad (9)$$

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

$$\Delta\xi = \frac{v \cdot \Delta t \cdot \tan\phi}{L}. \qquad (10)$$

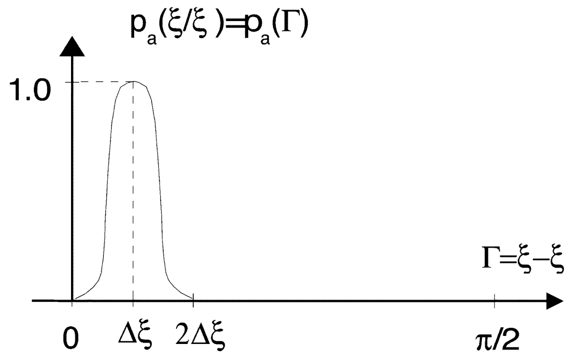


Fig. 15. Model of $\mathrm{p_a}(\xi \,|\, \xi')$.

Although $\Delta\xi$ cannot be regarded as an exact value, due to sliding, backlash, and measure noise not explicitly considered in the model, but (10) is helpful in modeling $p_a(\xi \,|\, \xi')$. The probability that the vehicle reaches position ξ, from the previous one at ξ', can be assumed to be the same as the probability that it covers an angular trajectory $\psi = \xi - \xi'$ in a time interval of Δt. Based on the previously described model, $\Delta\xi$ is the most likely value for $p_a(\xi \,|\, \xi')$. Likewise, the probability will gradually diminish as the difference between ξ and ξ' increases. On the other hand, the vehicle's forward movement is physically constrained, making it highly improbable that its angular position decreases with time

$$p_a(\xi \,|\, \xi') = 0 \qquad \forall \xi < \xi'. \tag{11}$$

Following this reasoning, we propose to model $p_a(\xi \,|\, \xi')$ using an auxiliary random variable denoted by Γ. Probability $p_a(\xi \,|\, \xi')$ can be substituted by probability $p_a(\Gamma = \psi)$, so that $p_a(\Gamma < 0) = 0$, while a Gaussian function is used for positive values of Γ, yielding a maximum at $\Gamma = \Delta\xi$. The model for $p_a(\xi \,|\, \xi')$ is graphically depicted in Fig. 15. Thus, the model can be analytically represented by a truncated Gaussian function in the sense that it is zero for negative values.

The vehicle kinematic model is not a completely precise way to determine the increment in the angular location of the vehicle between two consecutive iterations. In practice, it has been observed how the standard deviation of the Gaussian model is approximately equal to the average value $(\Delta\xi)$. Accordingly, the standard deviation of the Gaussian model for $p_a(\xi \,|\, \xi')$ has been empirically set to $\sigma = \Delta\xi$, so that the probability of angular values near $\Gamma = 0$ is almost zero. However, due to the fact that $p_a(\eta = \xi)$ is zero for negative values of Γ, the proposed model is not exactly a Gaussian function. This is corrected by using a normalizing factor $p(s)$, as described in the next section.

2) $\mathrm{Bel}(\xi)$ *Updating Upon Image Acquisition:* $\mathrm{Bel}(\xi)$ distribution must be validated according to the visual information contained in the scene acquired by the vehicle's vision system. Let s denote the vision system measurement, representing the degree of similarity or correlation, on a pixel-by-pixel basis, between the current segmented image and the *a priori* binary road models expected to be perceived upon intersection completion. A detailed definition of visual measurement s is provided later in this section. On the other hand, $p(s \,|\, \xi)$ represents the probability of obtaining measure s at position ξ. The belief distribution $\mathrm{Bel}(\xi)$ is updated upon image acquisition and processed according to

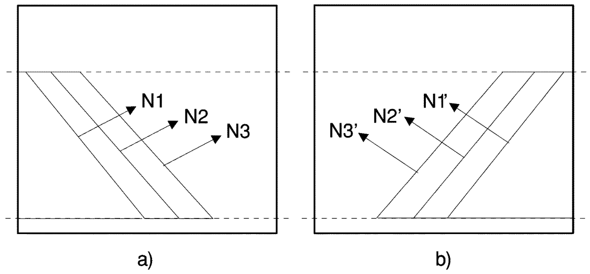


Fig. 16. *A priori* road models for (a) left and (b) right turns at intersections.

$$\mathrm{Bel}(\xi_t = \xi) = \frac{p(s \,|\, \xi) \cdot \widehat{\mathrm{Bel}}(\xi_t = \xi)}{p(s)} \tag{12}$$

where $p(s)$ represents a normalizing factor to ensure that $\mathrm{Bel}(\xi)$ is truly a real pdf. Visual measure s is obtained by computing, on a pixel-by-pixel basis, the correlation between the incoming segmentation and several *a priori* road models. For this purpose, three *a priori* road models have been devised for each turning direction, three for left turns and another three for right turns. The models, denoted by N_1, N_2, N_3 for left turns and N_1', N_2', N_3' for right turns, are located on the area of the image, where the road is most likely to first appear after the turn at an intersection has been completed. In addition, the width of the *a priori* road models is randomly chosen in a given interval from the road width $\hat{W}(t_0)$, estimated just before the intersection maneuver starts. This enables the system to recognize and track roads with different widths. Fig. 16 shows the shape of the *a priori* road models for left and right turns.

For each *a priori* road model, coefficients r_i and $\overline{r_i}$ are computed. Coefficient r_i measures the similarity between the road area in the segmented image and the road area in *a priori* model i. Likewise, $\overline{r_i}$ measures the correlation between the nonroad area in the segmented image and the no road area in *a priori* model i. These coefficients are calculated as shown in

$$\begin{aligned} r_i &= \frac{N_{i-\mathrm{road}}}{T_{i-\mathrm{road}}} \\ \overline{r_i} &= \frac{N_{i-\mathrm{noroad}}}{T_{i-\mathrm{noroad}}} \end{aligned} \tag{13}$$

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

$$s = \max_i \frac{r_i + \overline{r_i}}{2} \tag{14}$$

where s is in the range $0 \le s \le 1$. The modeling of conditional probability $p_a(s \,|\, \xi)$ is accomplished, taking into account the dynamic constraints on the vehicle's steering system. In order to anticipate the trajectory and to avoid overshoot and oscilla-

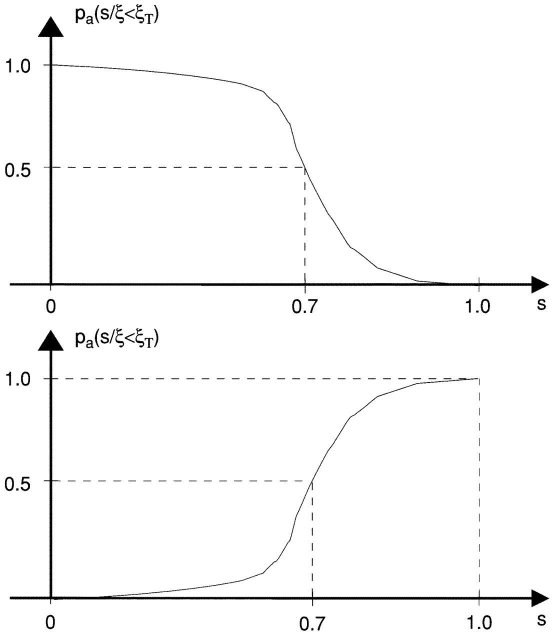


Fig. 17. Modeling of $\mathrm{p_a}(s \mid \xi)$.

tions, the recovery turning maneuver should start a little before the turn is completely finished. Accordingly, an experimental value $\xi_T = 70°$ is established to indicate that the probability of starting the recovery turn maneuver greatly increases upon completing angular trajectory $\xi_T \geq 70°$, whenever the correlation measure s validates the estimation. The system is mainly conceived to navigate in T-cross intersections in which the vehicle has to carry out 90° curves most of the time. For this reason, a value $\xi_T = 70°$ is a minimum sufficient value to ensure that the vehicle is close to finishing the curve.

An accurate environment map should be used to model $p_a(s \mid \xi)$ precisely. In earlier research on mobile robot localization in indoor environments [47], probability $p_a(s \mid l)$ is precomputed (where l represents the robot's location) based on a global map and a sensor model and is stored on a lookup table. After using lookup table, online computation of $p_a(s \mid l)$ is a quick and simple process. In reduced environments, this kind of technique is employed for radar- or laser-based systems. Considering that none of the previous conditions occur in a vision-based system in large outdoor scenarios, the use of precomputation becomes quite a complex and inefficient task. We propose instead a simple and intuitive modeling, successfully tested in practice, by which the probability of measuring a high value of s is very low at the beginning of the turn, but becomes increasingly higher as the angular trajectory of the vehicle gradually approaches ξ_T. Fig. 17 depicts the exact model for $p_a(s \mid \xi)$.

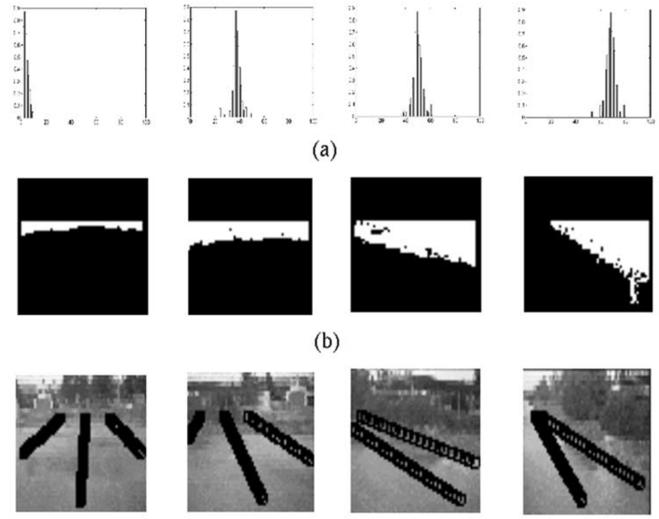
Modeling of probability $p_a(s \mid \xi)$ has been split into two intervals. For any value smaller than ξ_T, the probability of measuring a high correlation s is low, while the probability of obtaining a low correlation gradually increases. For angles greater than ξ_T, when the vehicle is close to completing the turn, the probability of measuring a high correlation s increases, maintaining an otherwise low value. Although $p_a(s \mid \xi)$ is not a real pdf (the integral of $p_a(s \mid \xi)$ along its definition domain does not add up to 1.0), we can be sure that $\mathrm{Bel}(\xi)$ distribution is a real pdf due to the normalizing factor $p(s)$ in (12). Considering that vehicle movement and image acquisition are carried out continuously and simultaneously, the $\mathrm{Bel}(\xi)$ distribution is updated during each iteration of the algorithm by applying (6) and (12) consecutively. From the practical point of view, the definition domain of variable ξ must be discretized so as to make the problem computationally treatable. An angular resolution of 0.5°, providing a more than sufficiently precise approximation of the location, has been set for this purpose. This implies that, for a typical angular range in a turn of about 90°, the number of probabilities to compute amounts to 180 × 180. However, most of the time probabilities other than zero are focused on a narrow interval. This means that a selective computation that increases the algorithm execution speed can be accomplished by considering only those angular values of ξ for which the probability $\mathrm{Bel}(\Upsilon = \xi)$ is above a given threshold (1% of the maximum probability in this case). The Markov localization method described in this section provides a belief distribution $\mathrm{Bel}(\xi)$ that tends to have a Gaussian shape. The average of this approximately Gaussian $\mathrm{Bel}(\xi)$ represents the most probable location of the vehicle. In order to obtain a measure of the reliability of the estimated location of the vehicle, the belief distribution $\mathrm{Bel}(\xi)$ is compared to an ideal Gaussian function $N(\xi_{\max}, \Delta\xi)$ (where $\xi_{\max}$ is the average of the Gaussian function that best fits $\mathrm{Bel}(\xi)$). The comparison is performed in the least-squares sense, as in

$$\Xi = \frac{1}{N_\xi} \sum_\xi \left(\mathrm{Bel}(\xi) - \frac{1}{\sqrt{2\pi}\Delta\xi} \exp^{\frac{-(\xi-\xi_{\max})^2}{2\cdot\Delta\xi^2}} \right)^2 \qquad (15)$$

where Ξ represents the mean-square error between distribution $\mathrm{Bel}(\xi)$ and the Gaussian function $N(\xi_{\max}, \Delta\xi)$ and N_ξ is the number of points of the discrete definition domain of variable ξ. The vehicle should finish the turning maneuver and resume lane tracking when the estimated angular position $\xi_{\max}$ is above ξ_T (i.e., close to the end of the turn) and, simultaneously, Ξ is below a given threshold (experimentally set to 2.5), indicating high confidence in the vehicle's estimated location. It was empirically observed that in practice the mean-square error of the belief distribution was bellow 2.5 in more than 95% of the cases in which the segmentation was appropriate. Accordingly, this value was chosen as a constraint that must be satisfied in order to consider that a maneuver at an intersection is about to finish. The intersection maneuver can be stopped at any point in which enough road is visible from the camera, as long as the lane-tracking algorithm is able to regain control of the vehicle from that point onward. If the square error is bellow 2.5, the segmentation will be most certainly poor, surely caused by not having enough visibility of the road. To sum up, the Markov localization method allows visual measures to be statistically enhanced and its ability to manage uncertainty and degrees of reliability makes it ideal for vehicle localization.

C. Intersection Navigation Results

To illustrate the behavior of the navigation system described in this section, Fig. 18 depicts a sequence of four real images of a left-turn maneuver at an intersection. In Fig. 18, we represent the



(a)

(b)

(c)

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

(d)

Fig. 18. Sequence of images of a left-turn maneuver. (a) Belief distribution, (b) segmentation, (c) estimated road model, and (d) localization results.

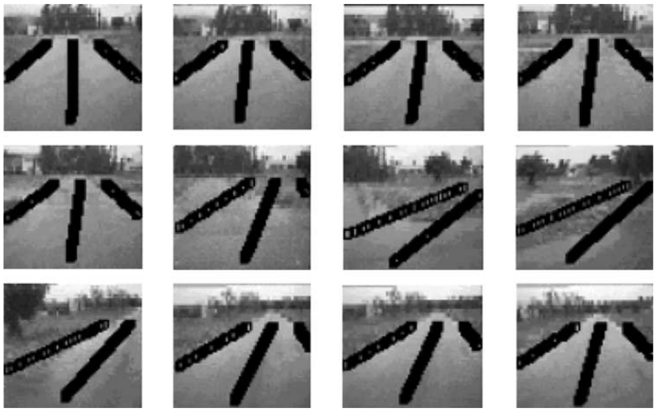


Fig. 19. Estimated road model for a complete concatenation of actions: lane tracking, intersection navigation, lane tracking.

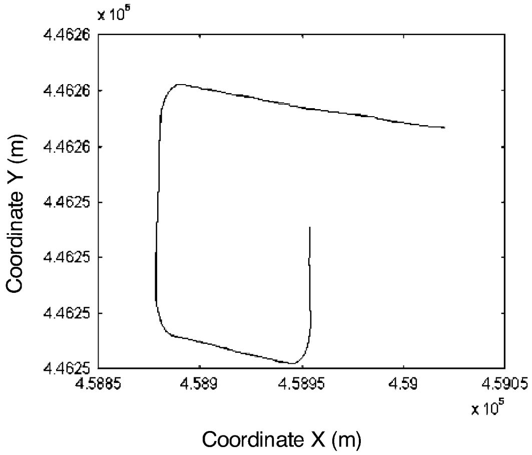


Fig. 20. Real trajectory followed by the vehicle on an autonomous mission between stations 1 and 2.

belief distribution for each image, the segmented image, the estimated road model that overprints the original incoming scene and the values of $s, \xi_{\max}$, and Ξ. As can be seen in Fig. 18, the vehicle's knowledge of its position throughout the entire turning maneuver remains reliably high (i.e., a low value of Ξ). It can also be seen how the road model begins at the last location estimated during lane tracking, just before commencing the turn (the first image of the sequence in Fig. 18) and gradually updates until it reaches the fixed *a priori* road model devised for left turns (the third image of the sequence in Fig. 18). When conditions for completing the turn are met ($\xi_{\max} > \xi_T$ and $\Xi < 2.5$), lane tracking is resumed, allowing the road model to be adapted to the new road, as shown in the last image of the sequence in Fig. 18. To gain a better understanding of the global process, Fig. 19 shows a complete example in which, from images 1–5, the vehicle tracks a lane until it reaches an intersection. In images 6–9, it performs a right-turn maneuver and, finally, lane tracking is resumed in images 10–12. To sum up, the following points must be made. The navigation module proposed in this section provides continuity during the road-model estimation and ensures accurate maneuvering at intersections of arbitrary angular shape, using only one color camera. The localization method can detect roads with different widths after completing the turn at an intersection.

V. Implementation and Results

The complete navigation system described in the previous sections has been implemented on the Babieca prototype vehicle, a commercial electric Citroen Berlingo, depicted in Fig. 1, which has been modified to allow for automatic velocity and steering control at a maximum speed of 90 km/h. The Babieca is equipped with a color camera, a DGPS receiver, a pentium PC, and a set of electronic devices to control the accelerator and steering wheel and to encode the vehicle's velocity and steering angle. The color camera provides a standard phase-altering lines (PAL) video signal at 25 Hz that is processed by a Meteor frame grabber installed on a 120-MHz pentium PC running a real-time Linux operating system. The DGPS receiver is a Z-12 real-time model by Ashtech that implements the RTCM SC 104 V2.2 standard at 5 Hz. Both velocity (longitudinal) and steering (lateral) control have been implemented in this work in order to provide completely autonomous operation. The longitudinal control module enables the vehicle to maintain the reference velocity established in the global velocity profile, computed at the beginning of the autonomous mission. For this purpose, a simple but robust fuzzy controller [14] has been designed. The main aim of the lateral control module is to ensure proper tracking of the road by correctly keeping the vehicle in the center of the lane and moving in the right direction (parallel to the road trajectory). The description of the lateral controller is outside the scope of this paper and can be found in [43]. Further theoretical support to better understand the development of the lateral controller can be found in [12] and [41].

The complete navigation system is implemented on real-time Linux using a preemptive scheduler [2]. The lane-tracking vision based task is executed at 10 frames/s, while intersection navigation is run at 4–5 frames/s. Practical experiments were

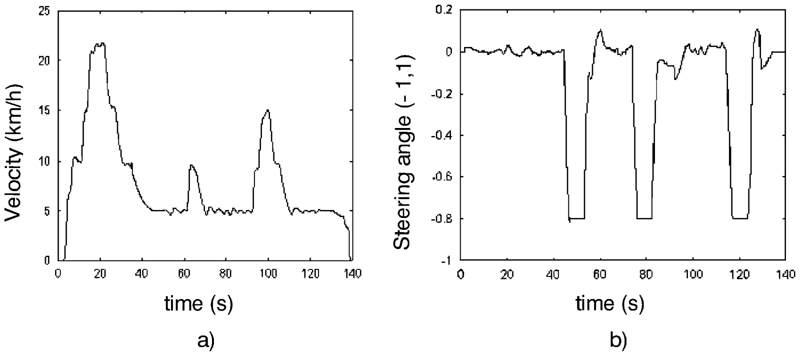


Fig. 21. Autonomous mission from station 1 to station 2. (a) Vehicle velocity and (b) vehicle-steering angle.

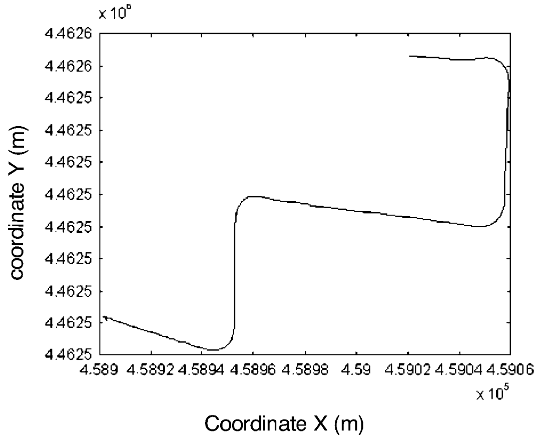


Fig. 22. Real trajectory followed by the vehicle on an autonomous mission between stations 5 and 1.

carried out on a private circuit at the Industrial Automation Institute, Arganda del Rey, Madrid. The circuit, shown in Fig. 2, is designed to emulate an urban quarter with several stopping points, streets, intersections, and roundabouts. Babieca ran for hundreds of kilometers on many successful autonomous missions over this circuit.

To illustrate the global behavior of the complete navigation system implemented on Babieca, some general results are shown next. In the first test, the vehicle was commanded to autonomously navigate from station 1 to station 2. Fig. 20 shows the two-dimensional (2–D) real trajectory followed by Babieca using universal transverse mercator (UTM) coordinates. Likewise, the vehicle's real velocity and steering angle during the mission are depicted in Fig. 21, clearly showing the strong turns performed at intersections. Similarly, Fig. 22 shows the global trajectory covered by the vehicle in getting from station 5 to station 1.

A live demonstration exhibiting the system's capabilities at autonomous navigation was carried out during the IEEE Conference on Intelligent Vehicles 2002, on a private circuit in Satory, Versailles, France. In order to complete the graphical results depicted in this section, a complete set of video files demonstrating the operational performance of the system in real tests can be retrieved from ftp://www.depeca.uah.es/pub/vision.

VI. Conclusion and Future Work

The main novelty of this work is the development of a vision- and DGPS-based global-navigation system capable of executing autonomous missions on a network of unmarked roads and intersections. It currently is implemented on a slightly modified commercial vehicle. The complete system has been successfully tested on a private circuit, as a first step toward its long-term deployment in urban scenarios. Depending on the specifications of the mission and the *a priori* map, the global-navigation system simultaneously implements two complementary vision-based behaviors for road tracking and navigation at intersections. A task manager correctly synchronizes the execution of the adequate vision-based task, depending on whether the vehicle is moving along a road or traversing an intersection, making use of the DGPS for this purpose. The fact that the DGPS signal does not have to be very precise, together with the use of a single color camera, results in a low-cost system that is suitable for midterm commercial development. Another major contribution of this work is the proper tracking of nonstructured roads, as it is robust, does not require previous instruction, and allows for real-time operation. Vision-based intersection navigation is another remarkable feature due to the complexity of the maneuver and the need to maintain continuity while navigating a network of roads. Nonetheless, a lot of work remains to be done before a truly robust and reliable autonomous system can be fully deployed in real conditions. In the next step, the already-existing vehicle detection module will be improved by combining information provided by other laser- or radar-based sensors. Another key point will be eradicating the dependence on DGPS by implementing a vision-based task for intersection detection, which uses a conventional GPS receiver. Finally, another vision-based specialized task will be developed in the future, which will be capable of navigating both intersections and roundabouts, as at the present time they are in widespread use in urban environments.

References

[1] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert, "Multi-robot cooperation in the MARTHA project," *IEEE Robot. Automat. Mag.*, vol. 5, pp. 36–47, Mar. 1998.

[2] M. Barabanov, "A Linux-based real-time operating system," M.S. thesis, New Mexico Instit. Mining Technology, Dept. Comput. Sci., Socorro, NM, June 1997.

[3] M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Processing*, vol. 7, pp. 62–81, Jan. 1998.

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

[5] A. Broggi and S. Bert, "Vision-based road detection in automotive systems: A real-time expectation-driven approach," *J. Artif. Intell. Res.*, vol. 3, pp. 325–348, 1995.

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

[7] A. Broggi, S. Nichele, M. Bertozzi, and A. Fascioli, "Stereo vision-based vehicle detection," in *Proc. IEEE Intelligent Vehicles Symp.*, Detroit, MI, Oct. 2000, pp. 39–44.

[8] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, "Shape-Based Pedestrian Detection," in *Proc. IEEE Intelligent Vehicles Symp.*, Detroit, MI, Oct. 2000, pp. 215–22.

[9] S. Cameron and P. Proberdt, *Advanced Guided Vehicles. Aspects of the Oxford AGV Project*. Singapore: World Scientific, 1994.

[10] R. Chapuis, R. Aufrre, and F. Chausse, "Recovering the 3D shape of a road by vision," in *Proc. Int. Conf. Image Processing Applications*, Manchester, U.K., July 1999, pp. 686–690.

[11] P. Charbonnier, P. Nicolle, Y. Guillard, and J. Charrier, "Road boundaries detection using color saturation," in *Proc. 9th Eur. Signal Processing Conf.'98*, Sept. 1998, pp. 2553–2556.

[12] L. Cordesses, P. Martinet, B. Thuilot, and M. Berducat, "GPS-based control of a land vehicle," in *Proc. IAARC/IFAC/IEEE Int. Symp. Automation Robotics Construction (ISARC)'99*, Madrid, Spain, Sept. 22–24, 1999, pp. 41–46.

[13] J. D. Crisman and C. E. Thorpe, "UNSCARF, a color vision system for the detection of unstructured roads," in *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, 1991, pp. 2496–2501.

[14] T. De Pedro, R. Garcia, C. Gonzalez, J. E. Naranjo, J. Reviejo, and M. A. Sotelo, "Vehicle automatic driving system based on GNSS," in *Proc. Int. Conf. Intelligent Vehicles*, Seville, Spain, 2001.

[15] S. Denasi, C. Lanzone, P. Martinese, G. Pettiti, G. Quaglia, and L. Viglione, "Real-time system for road following and obstacle detection," in *Proc. SPIE Machine Vision Applications, Architectures, Systems Integration III*, 1994, pp. 70–79.

[16] E. D. Dickmanns and A. Zapp, "A curvature-based scheme for improving road vehicle guidance by computer vision," in *Proc. SPIE Mobile Robots*, vol. 727, 1986, pp. 161–168.

[17] E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrant, M. Mauer, F. Thomanek, and J. Shielhlen, "The seeing passenger car "VaMoRs-P"," in *Proc. Int. Symp. Intelligent Vehicles*, Paris, France, 1994, pp. 68–73.

[18] E. D. Dickmanns, "Vehicles capable of dynamic vision," in *Proc. 15th Int. Joint Conf. Artificial Intelligence (IJCA)'97*, Nagoya, Japan, Aug. 23–29, 1997, pp. 1577–1592.

[19] U. Franke, D. Gavrila, S. Gorzig, F. Lindner, F. Paitzold, and C. Wohler, "Autonomous driving goes downtown," in *Proc. IEEE Intelligent Vehicles Symp.*, Stuttgart, Germany, 1998, pp. 40–48.

[20] A. Giachetti, M. Campani, and V. Torre, "The use of optical flow for road navigation," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 34–48, Feb. 1998.

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

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

[23] N. Ikonomakis, K. N. Plataniotis, and A. N. Venetsanopoulos, "Color image segmentation for multimedia applications," *J. Intell. Robot. Syst.*, pp. 5–20, 2000.

[24] W. D. Jones, "Keeping cars from crashing," *IEEE Spectr.*, vol. 38, pp. 40–45, Sept. 2001.

[25] K. J. Kim, S. Y. Oh, S. W. Kim, H. Jeong, C. N. Lee, B. S. Kim, and C. S. Kim, "An autonomous land vehicle PRV II: Progresses and performance enhancement," in *Proc. IEEE Intelligent Vehicles Symp.*, Detroit, MI, 1995, pp. 264–269.

[26] K. Kluge and C. Thorpe, "Intersection detection in the YARF road following system," in *Ingelligent Autonomous Systems*. Pittsburgh, PA: Kluwer, 1993, vol. 3.

[27] W. Li, X. Jiang, and Y. Wang, "Road recognition for vision navigation of an autonomous vehicle by fuzzy reasoning," in *Fuzzy Sets and Systems*. Amsterdam, The Netherlands: Elsevier, 1998, vol. 93.

[28] M. Lutzeler and E. D. Dickmanns, "Road recognition with MarVEye," in *Proc. IEEE Intelligent Vehicles Symp.*, Stuttgart, Germany, 1998, pp. 112–117.

[29] R. Gregor, M. Lutzeler, M. Pellkofer, K. H. Siedersberger, and E. D. Dickmanns, "EMS-vision: A perceptual system for autonomous system," *IEEE Trans. Intell. Transport. Syst.*, vol. 3, pp. 48–59>, Mar. 2002.

[30] H. P. Moraveck, "Toward automatic visual obstacle avoidance," in *Proc. 5th Int. Joint Conf. Artificial Intelligence*, Aug. 1977, pp. 584–589.

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

[32] D. A. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*. Norwell, MA: Kluwer, 1993.

[33] D. A. Pomerleau and T. M. Jockem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Expert*, vol. 11, pp. 19–27, Apr. 1996.

[34] K. A. Redmill, "A simple vision system for lane keeping," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, Boston, MA, 1997, pp. 212–217.

[35] J. Reviejo, T. de Pedro, J. E. Naranjo, R. Garcia, C. Gonzalez, and S. Alcalde, "An open frame to test techniques and equipment for autonomous vehicles," in *Proc. IFAC Symp. Manufacturing, Modeling, Management and Control*, Rio, Greece, July 2000, pp. 350–354.

[36] R. Risack, P. Klausmann, W. Kruger, and W. Enkelmann, "Robust lane recongnition embedded in a real-time driver assistance system," in *Proc. IEEE Intelligent Vehicles Symp.*, Stuttgart, Germany, 1998, pp. 35–40.

[37] F. J. Rodriguez, M. Mazo, and M. A. Sotelo, "Automation of an industrial fork lift truck, guided by artificial vision in open environments," in *Autonomous Robots*. Norwell, MA: Kluwer, 1998, vol. 5, pp. 215–231.

[38] S. Rossle, V. Kruger, and G. Gengenbach, "Real-time vision-based intersection detection for a driver's warning assistant," in *Proc. Intelligent Vehicle Conf.'93*, Tokyo, Japan, 1993, pp. 340–344.

[39] H. Schneiderman and M. Nashman, "A discriminating feature tracker for vision-based autonomous driving," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 769–775, Dec. 1994.

[40] H. Schneiderman, A. J. Wavering, M. Nashman, and R. Lumia, "Real time model-based visual tracking," in *Proc. Int. Workshop Intelligent Robotic Systems*, 1994.

[41] J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[42] M. A. Sotelo, A. Zelinsky, F. J. Rodriguez, and L. M. Bergasa, "Real-time road tracking using templates matching," in *Proc. 3rd Int. Symp. Intelligent Automation (IIA)'99 and Soft Computing (SOCO)'99*, Genoa, Italy, 1999.

[43] M. A. Sotelo, "Sistema de Navegación Global Aplicado al Guiado de un Vehiculo Autónomo Terrestre en Entornos Exteriores Parcialmente Conocidos," Ph.D. Dissertation, Univ. Alcala, Madrid, Spain, Oct. 2001.

[44] A. Stentz and M. Hebert, "A complete navigation system for goal acquisition in unknown environments," in *Autonomous Robots*. Norwell, MA: Kluwer, 1995, vol. 2, pp. 127–145.

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

[46] C. Thorpe, *Vision and Navigation: The Carnegie Mellon Navlab*. Norwell, MA: Kluwer, 1990.

[47] D. F. W. Burgard and S. Thrun, *Active Markov Localization for Mobile Robots*. Amsterdam, The Netherlands: Elsevier, Nov. 1998.

[48] L. Zhao and C. Thorpe, "Stereo- and neural network-based pedestrian detection," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems'99*, Tokyo, Japan, Oct. 1999, pp. 298–303.

Miguel Angel Sotelo (M'02) received the Dr.Ing. degree in telecommunication engineering from the Technical University of Madrid, Madrid, Spain, in 1996, and the Ph.D. degree in telecommunication engineering from the University of Alcalá, Alcalá de Henares, Madrid, Spain, in 2001.

From 1993 to 1994, he was a Researcher with the Department of Electronics, University of Alcalá, where he currently is an Associate Professor. He is the author of more than 60 refereed publications in international journals, book chapters, and conference proceedings. His research interests include real-time computer vision and control systems for autonomous and assisted intelligent vehicles, as well as vision and DGPS-based global navigation systems.

Dr. Sotelo was the Recipient of the Best Research Award in the domain of Automotive and Vehicle Applications in Spain, in 2002.

Francisco Javier Rodriguez (S'99–M'00) received the Ph.D. degree in electronics engineering from the University of Alcalá, Alcalá de Henares, Madrid, Spain, in 1997, the Telecommunication degree from the Technical University of Madrid, Madrid, in 1990, and the Technical Telecommunication Engineering degree from the University of Alcalá in 1985.

He has worked in the private electronic industry for two years. Since 1986, he has been a Lecturer in the Electronics Department, University of Alcalá, where he is an Associate Professor. His research interests include the areas of monitoring and control, real-time processing, autonomous vehicles, and embedded systems.

Luis Magdalena (S'89–M'92) received the M.S. and Ph.D. degrees in telecommunication engineering from the Technical University of Madrid, Madrid, Spain, in 1988 and 1994, respectively.

He is an Associate Professor of Computer Science with the Department of Applied Mathematics, Technical University of Madrid. He is coauthor of *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* (Singapore: World Scientific, 2001) and is Coeditor of *Technologies for Constructing Intelligent Systems* (two volumes, Phisica Verlag, Heidelberg, Germany: 2002) and *Fuzzy Modeling and the Interpretability-Accuracy Trade-Off* (two volumes, Phisica Verlag, Heidelberg, Germany: 2002). His research interests include soft computing and its application, particularly in the field of robotics.

Dr. Magdalena has coedited special issues of the journals *International Journal of Intelligent Systems*, *Information Sciences*, and *Fuzzy Sets and Systems*. Since 2001, he has been President of the European Society for Fuzzy Logic and Technology. He served as Cochair of the Eighth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Madrid, 2000.