Autonomous Navigation of Vehicles from a Visual Memory Using a Generic Camera Model

Jonathan Courbon, Youcef Mezouar, and Philippe Martinet

***Abstract*—In this paper, we present a complete framework for autonomous vehicle navigation using a single camera and natural landmarks. When navigating in an unknown environment for the first time, usual behavior consists of memorizing some key views along the performed path to use these references as checkpoints for future navigation missions. The navigation framework for the wheeled vehicles presented in this paper is based on this assumption. During a human-guided learning step, the vehicle performs paths that are sampled and stored as a set of ordered key images, as acquired by an embedded camera. The visual paths are topologically organized, providing a visual memory of the environment. Given an image of the visual memory as a target, the vehicle navigation mission is defined as a concatenation of visual path subsets called visual routes. When autonomously running, the control guides the vehicle along the reference visual route without explicitly planning any trajectory. The control consists of a vision-based control law that is adapted to the nonholonomic constraint. Our navigation framework has been designed for a generic class of cameras (including conventional, catadioptric, and fisheye cameras). Experiments with an urban electric vehicle navigating in an outdoor environment have been carried out with a fisheye camera along a 750-m-long trajectory. Results validate our approach.**

***Index Terms*—Autonomous navigation, generic camera model, monocular vision, nonholonomic mobile vehicle, real-time application, robot navigation, urban vehicles, visual memory.**

I. Introduction

THE saturation of vehicle traffic in large cities is a major concern. Improvements can be gained from the development of alternative public transportation systems. To meet public expectations, such systems should be very flexible so that they are suitable to answer many different individual needs and be as nuisance free as possible (with respect to pollution, noise, urban scenery, etc.). Individual electric vehicles, which are available in a car-sharing concept, clearly meet both requirements. They appear to be very suitable in specific areas where the public demand is properly structured, such as in airport terminals, attraction resorts, university campuses, or inner-city pedestrian zones. To develop such a transportation system, the automatic navigation of those vehicles has to be addressed. Passengers could then move from any point to any other point at their convenience in an automatic way, and vehicles could autonomously be brought back to stations for refueling and reuse.

Such an automatic navigation may be obtained with visual sensors. The authors of [1] account for 20 years of work at the intersection between robotics and computer vision communities. In many works, as in [2], computer vision techniques are used in a landmark-based framework. Identifying the extracted landmarks with known reference points allows to update the results of the localization algorithm. These methods are based on some knowledge about the environment, such as a given 3-D model or a map built online. They generally rely on a complete or partial 3-D reconstruction of the observed environment through the analysis of data collected from disparate sensors. The vehicle can thus be localized in an absolute reference frame. Both motion planning and vehicle control can then be designed in this space. The results obtained by the authors of [3] leave to be forecasted that such a framework will be reachable using a single camera. However, although an accurate global localization is unquestionably useful, our aim is to build a complete vision-based framework without recovering the position of the vehicle with respect to a reference frame (in [1], this type of framework is ranked among qualitative approaches) and suitable for a large class of vision sensors.

Method Overview and Paper Structure

An overview of the proposed navigation framework is presented in Fig. 1. The method can be divided into the following three steps: 1) visual memory building, 2) localization, and 3) autonomous navigation.

In the first offline step (visual memory building), a sequence of images is acquired during a human-guided navigation. It allows us to derive paths that drive the vehicle from its initial locations to its goal locations. To reduce the complexity of the image sequences, only key views are stored and indexed on a visual path. The set of visual paths can be interpreted as a visual memory of the environment. Section II more precisely details this point.

Manuscript received February 10, 2008; revised September 30, 2008. First published February 10, 2009; current version published September 1, 2009. The Associate Editor for this paper was U. Nunes.

J. Courbon is with the Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique (LASMEA), 63177 Aubiere, France, and also with Centre dEtude Atomique (CEA), List, 92265 Fontenay Aux Roses, France (e-mail: jonathan.courbon@lasmea.univ-bpclermont.fr).

Y. Mezouar is with the Blaise Pascal University, 63006 Clermont-Ferrand, France, and also with the Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique (LASMEA), 63177 Aubiere, France (e-mail: youcef.mezouar@lasmea.univ-bpclermont.fr).

P. Martinet is with the Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique (LASMEA), 63177 Aubiere, France, and also with the Institut Franćais de Mécanique Avancée (IFMA), 63175 Aubiere, France (e-mail: philippe.martinet@lasmea.univ-bpclermont.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2008.2012375

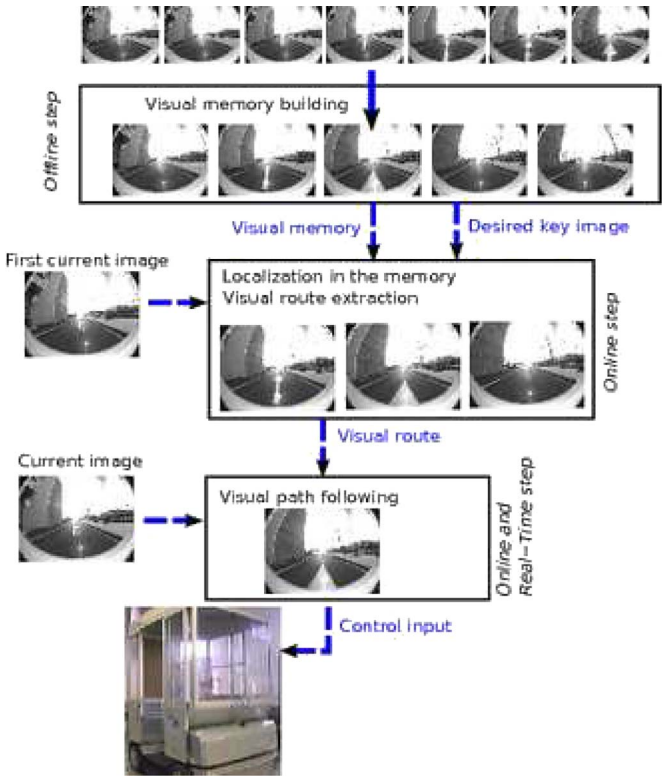


Fig. 1. Overview of the proposed framework.

In the second step, before the beginning of the motion, the localization of the robotic system is performed. During this stage, no assumption about the vehicle's position is made. The localization process consists of finding the image that best fits the current image in the visual memory. With this aim, we propose a hierarchical process that combines the global descriptors computed by cubic interpolation of a triangular mesh and patch correlation around Harris corners. Note that we only seek the most similar view and not the metric position of the robotic system. More details about this process are given in Section III.

In the last stage (refer to Section IV), given an image of one of the visual paths as a target, the vehicle navigation mission is defined as a concatenation of visual path subsets called visual route. A navigation task then consists of autonomously executing a visual route. The vehicle is controlled by a vision-based control law that is adapted to its nonholonomic constraint. This control guides the vehicle along the reference visual route without explicitly planning any trajectory. Note that in our approach the control part takes into account the model of the vehicle.

Experiments have been carried out with an electrical urban vehicle, navigating in an outdoor environment along a 750-m-long trajectory. The results are presented in Section V.

II. Visual Memory and Routes

In [1], approaches that use a "memorization" of images of the environment acquired with an embedded camera are ranked among mapless navigation systems. As proposed in [4] or [5], no notion of mapping nor topology of the environment appears in building the reference set of images, nor for the automatic guidance of the vehicle. The first step of our framework consists of a learning stage to build the visual memory. The visual memory is structured according to the environment topology to reduce the computational cost.

A. Visual Memory Structure

The learning stage relies on human experience. The user guides the vehicle along one or several paths into each place where the vehicle is authorized to go. A visual path Ψ^p is then stored and indexed as the pth learned path.

1) Visual Paths: A visual path Ψ^p is a weighted directed graph composed of n successive key images (*vertices*)

$$\Psi^p = \{\mathcal{I}_i^p \mid i \in \{1, 2, \ldots, n\}\}.$$

For control purposes (see Section IV), the authorized motions during the learning stage are assssumed to be limited to those of a car-like vehicle, which only goes forward. The following hypothesis formalizes these constraints.

Hypothesis 1: Given two frames ${}^R\mathcal{F}_i$ and ${}^R\mathcal{F}_{i+1}$, respectively, associated to the vehicle when two successive key images $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ of a visual path Ψ were acquired, there exists an admissible path ψ from ${}^R\mathcal{F}_i$ to ${}^R\mathcal{F}_{i+1}$ for a car-like vehicle whose turn radius is bounded and only moves forward.

Moreover, because the controller is vision based, the vehicle is controllable from $\mathcal{I}_i$ to $\mathcal{I}_{i+1}$ only if the hereunder Hypothesis 2 is respected.

Hypothesis 2: Two successive key images $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ contain a set $\mathcal{P}_i$ of matched visual features, which can be observed along a path performed between ${}^R\mathcal{F}_i$ and ${}^R\mathcal{F}_{i+1}$ and which allows the computation of the control law.

In the sequel, we use interest points as visual features. During the acquisition of a visual path, Hypothesis 2 constrains the choice of the key images. The key image selection process is detailed in Section II-C. As a consequence of Hypotheses 1 and 2, each visual path Ψ^p corresponds to an oriented edge that connects two configurations of the vehicle's workspace. The number of key images of a visual path is directly linked to the human-guided path complexity. The *weight of a visual path* is then defined as its cardinal.

2) Visual Memory Vertices: To connect two visual paths, the terminal extremity of one path and the initial extremity of the other path must be constrained as two consecutive key images of a visual path. The paths are then connected by a vertex, and two adjacent vertices of the visual memory are connected by a visual path.

Proposition 1: Given two visual paths $\Psi^{p_1} = \{\mathcal{I}_i^{p1} | i \in \{1, 2, \ldots, n_1\}\}$ and $\Psi^{p_2} = \{\mathcal{I}_i^{p2} | i \in \{1, 2, \ldots, n_2\}\}$, if the two key images $\mathcal{I}_{n_1}^{p_1}$ and $\mathcal{I}_1^{p_2}$ abide by both Hypotheses 1 and 2, then a vertex connects Ψ^{p_1} to Ψ^{p_2}.

We also assume Proposition 1 in the particular case where the terminal extremity of a visual path Ψ^{p_1} is the same key image as the initial extremity of another visual path Ψ^{p_2}. This is useful in practice when building the visual memory.

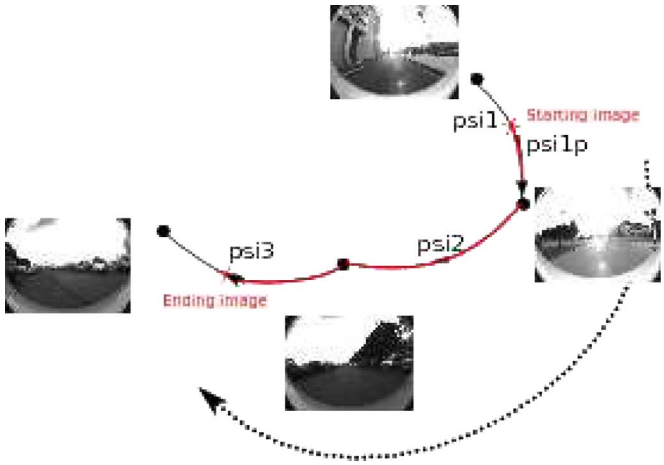


Fig. 2. Tasks consist of navigating from the starting to the ending images. With this aim, a visual route $\Xi = \Psi^{1'} \oplus \Psi^{2} \oplus \Psi^{3'}$ connecting these two images is defined.

3) Connected Multigraph of Weighted Directed Graphs: According to Sections II-A1 and 2, the visual memory structure is defined as a multigraph which vertices are key images linked by edges which are the visual paths (*directed graphs*). Note that more than one visual path may be incident to a node. It is necessary that this multigraph is strongly connected. This condition guarantees that any vertex of the visual memory is attainable from every other through a set of visual path.

B. Visual Route

A visual route describes the vehicle's mission in the sensor space. Given two key images of the visual memory $\mathcal{I}_s^*$ and $\mathcal{I}_g$, which correspond, respectively, to the starting and goal locations of the vehicle in the memory, a visual route is a set of key images that describes a path from $\mathcal{I}_s^*$ to $\mathcal{I}_g$, as presented in Fig. 2. $\mathcal{I}_s^*$ is the closest key image to the current image $\mathcal{I}_s$. The image $\mathcal{I}_s^*$ is extracted from the visual memory during a localization step, as described in Section III. The visual route is chosen as the minimum length path of the visual memory that connects the two vertices associated to $\mathcal{I}_s^*$ and $\mathcal{I}_g$. According to the definition of the value of a visual path, the length of a path is the sum of the values of its arcs. The minimum length path is obtained by using Dijkstra's algorithm. Consequently, the visual route results from the concatenation of indexed visual paths. Given two visual paths Ψ^{p_1} and Ψ^{p_2}, respectively, containing n_1 and n_2 indexed key images, the concatenation operation of Ψ^{p_1} and Ψ^{p_2} is defined as

$$\Psi^{p_1} \oplus \Psi^{p_2} = \left\{ \mathcal{I}_j^{p_{1,2}} | j = \{1, \ldots, n_1, n_1+1, \ldots, n_1+n_2\} \right\}$$

$$\mathcal{I}_j^{p_{1,2}} = \begin{cases} \mathcal{I}_j^{p_1}, & \text{if } j \leq n_1 \\ \mathcal{I}_{j-n_1}^{p_2}, & \text{if } n_1 < j \leq n_1 + n_2. \end{cases}$$

C. Key-Images Selection

A central clue for the implementation of our framework relies on efficient point matching. This process takes place in all steps of the proposed navigation framework. It allows key image selection during the learning stage; of course, it is also useful during autonomous navigation to provide the necessary input for state estimation (see Section IV). We use a process similar to that proposed in [3] and successfully applied for the metric localization of autonomous vehicles in an outdoor environment. Interest points are detected in each image with a Harris corner detector [6]. For an interest point $\mathcal{P}_1$ at coordinates (x, y) in image $\mathcal{I}_i$, we define a search region in image $\mathcal{I}_{i+1}$. The search region is a rectangle whose center has coordinates (x, y). For each interest point $\mathcal{P}_2$ inside the search region in image $\mathcal{I}_{i+1}$, we compute a score between the neighborhoods of $\mathcal{P}_1$ and $\mathcal{P}_2$ using a zero-normalized cross correlation (ZNCC). The point with the best score that is greater than a certain threshold is kept as a good match, and the unicity constraint is used to reject matches that have become impossible. This method is illumination invariant, and its computational cost is small.

The first image of the video sequence is selected as the first key frame $\mathcal{I}_1$. A key frame $\mathcal{I}_{i+1}$ is then chosen so that there are as many video frames as possible between $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$, whereas there are at least $\mathcal{M}$ common interest points tracked between $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$.

III. Localization in a Memory of Wide Field of View Images

The output of the learning process is a data set of images (*visual memory*). The first step of the autonomous navigation process is the self-localization of the vehicle in the visual memory. The localization consists of finding the image of the memory that best fits the current image by comparing pre-processed and online-acquired images. We particularly focus on a method that is suitable when the data set consists of omnidirectional images. Omnidirectional cameras are usually intended as a vision system that provides a huge field of view. Such an enhanced field of view can be achieved by either using catadioptric systems, which are obtained by opportunely combining mirrors and conventional cameras, or employing purely dioptric fisheye lenses [7]. As first demonstrated in [8] and exploited in robotic applications in [9], the images acquired by those sensors behave similarly. In our experiments, a fisheye camera is employed.

The efficiency of a visual localization method can be measured by means of the following: 1) accuracy of the results, 2) memory needed to store data, and 3) computational cost. Our main objective is to optimize the localization process under those criteria. Two main strategies exist to match the images: the image can be represented by a single descriptor (*global approaches*) [10]–[12] or alternatively by a set of descriptors defined around visual features (*landmark-based or local approaches*) [13]–[15]. In those last methods, some relevant visual features are extracted from the images. A descriptor is then associated to each feature neighborhood. The robustness of the extraction and the invariance of the descriptor are main issues to improve the matching process.

In one hand, local approaches are generally more accurate but have a high computational cost [15]. On the other hand,

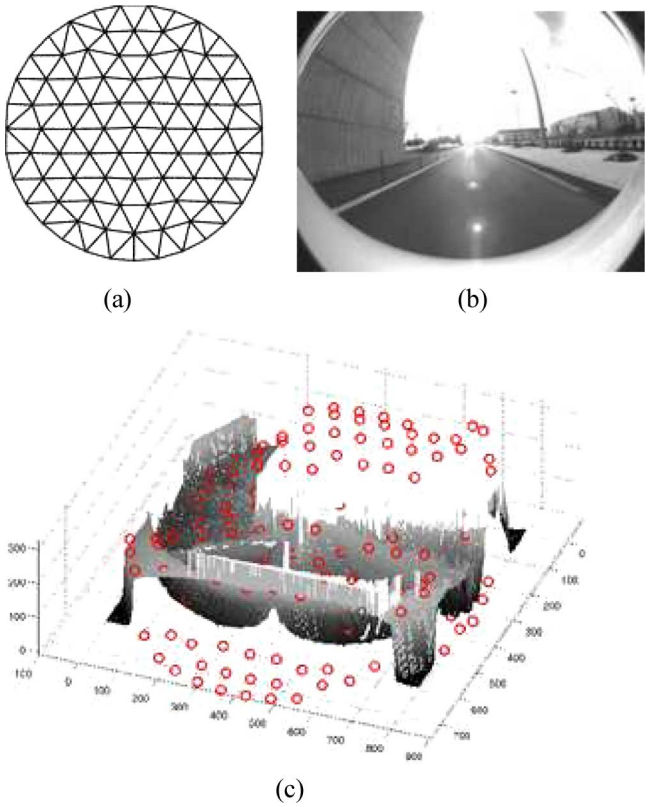


Fig. 3. Given the control point positions in the plane (a), the image (b) is seen as a surface and interpolated (c).

global descriptors speed up the matching process at the price of affecting the robustness to occlusions. Some approaches are hierarchical [15], [16]: A first selection is done by using a global descriptor, whereas the final localization results from local descriptors.

We propose a hierarchical approach for localization in a database of omnidirectional images. The computational efficiency is ensured in a first step by defining a well-suited global descriptor that allows to select a set of candidate images. Local descriptors are then exploited to only select the best image and thus to ensure accuracy.

A. Global Descriptor

The first step is based on a geometrical image representation that is derived from surface interpolation. Images first have their histogram equalized to be more robust to illumination changes. Pixels are seen as a discrete 3-D surface with the grey level as the third coordinates (see Fig. 3)

$$\mathcal{I} : \begin{cases} [0,1,\ldots,N] \times [0,1,\ldots,M] & \mapsto [0,255] \\ (u,v) & \to \mathcal{I}(u,v). \end{cases}$$

The interpolation consists of locally approximating this surface $\mathcal{I}(u,v)$ by a surface $f(s,t)$, $s \in [0;1], t \in [0;1]$. Note that it is necessary to have control points at the same positions to compare descriptors of different images. Moreover, regular positions ensure a better interpolation. In that aim, we propose to use the triangular mesh vertices represented in Fig. 3(a) as control points and the altitude Z of the control points of the approximated surface as descriptors. This triangular mesh is generated as proposed in [17]. It is based on the analogy between a simplex mesh and a truss structure. Meshpoints $\{P_1, P_2, \ldots P_p\}$ are nodes of the truss, and segments between two meshpoints are bars. An appropriate force-displacement function is applied to the bars at each iteration. This function takes into account the internal force due to the bars and the external force due to the boundaries. Node locations $P_i = (x_i y_i)$ are computed by solving for equilibrium in a truss structure using piecewise linear force–displacement relations.

The surface is interpolated by a cubic function. The required computational cost is low, and the interpolation errors are small.

B. First Selection and Local Descriptor

The descriptor Z_c (respectively, Z_i) is computed for the current image $\mathcal{I}_c$ (respectively, for the memorized image $\mathcal{I}_i$). The global distance between those two images is $d_i^{\text{global}} = \sum_{k=1}^{d} |Z_{c,k} - Z_{i,k}|$, where $Z_{i,k}$ corresponds to the kth element of the descriptor of the image $\mathcal{I}_i$. The kept candidate images are such that $d_i^{\text{global}} / \min_i d_i^{\text{global}} \leq t$, where the threshold $t \geq 1$ allows us to not reject the images that have a distance similar to the minimal distance. The output of this first stage is a small number of candidate images.

We then use a local approach to select the best candidate since only a few images are involved (i.e., in this case, the computational cost is low). With this aim, a classical local approach based on the ZNCC between patches around Harris corners is employed since the computational cost is much lower than scale-invariant feature transform- or speeded up robust features-based approaches, whereas a similar accuracy is obtained with images corresponding to close viewpoints (see [18] for detailed comparisons). In this stage, the local distance between two images is simply chosen as $d_i^{\text{local}} = 1/n$, where n is the number of matched features. The final result of the localization is the image $\mathcal{I}_k$ such that $d_k^{\text{local}} = \min_i(d_i^{\text{local}})$.

This hierarchical method has been compared to state-of-the-art techniques in [18]. The obtained results show that the proposed method is the best compromise between accuracy, amount of memorized data, and computational cost.

IV. Route Following

When starting the autonomous navigation task, the output of the localization step provides the closest image $\mathcal{I}_s^*$ to the current initial image $\mathcal{I}_c$. A visual route Ψ that connects $\mathcal{I}_s^*$ to the goal image is then extracted from the visual memory. As previously explained, the visual route is composed of a set of key images. The next step is to automatically follow this visual route using a visual servoing technique. The principle is presented in Fig. 4.

To design the controller, which is described in the sequel, the key images of the reference visual route are considered as consecutive checkpoints to reach the sensor space. The

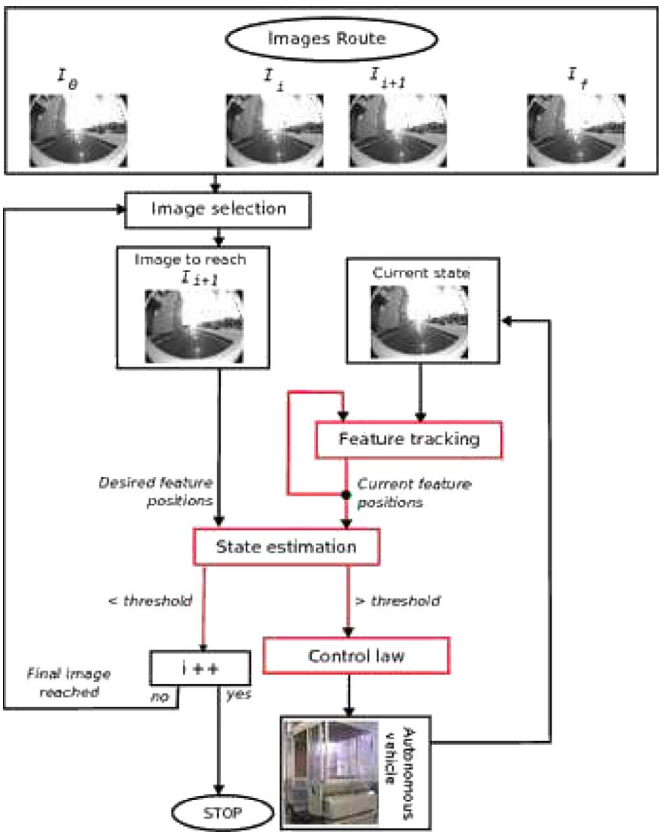


Fig. 4. Visual route-following process.

control problem is formulated as a path following to guide the nonholonomic vehicle along the visual route.

A. Model and Assumptions

1) Control Objective: Let $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ be two consecutive key images of a given visual route to follow, and let $\mathcal{I}_c$ be the current image. Let us note that $\mathcal{F}_i = (O_i, \mathbf{X}_i, \mathbf{Y}_i, \mathbf{Z}_i)$ and $\mathcal{F}_{i+1} = (O_{i+1}, \mathbf{X}_{i+1}, \mathbf{Y}_{i+1}, \mathbf{Z}_{i+1})$ are the frames attached to the vehicle when $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ were stored, and $\mathcal{F}_c = (O_c, \mathbf{X}_c, \mathbf{Y}_c, \mathbf{Z}_c)$ is the frame attached to the vehicle in its current location. Fig. 5 illustrates this setup. The origin O_c of $\mathcal{F}_c$ is on the center rear axle of a car-like vehicle, which moves on a perfect ground plane. The hand–eye parameters (i.e., the rigid transformation between $\mathcal{F}_c$ and the frame attached to the camera) are supposed to be known. According to Hypothesis 2, the state of a set of visual features $\mathcal{P}_i$ is known in the images $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$. The state of $\mathcal{P}_i$ is also assumed available in $\mathcal{I}_c$ (i.e., $\mathcal{P}_i$ is in the camera field of view). The task to achieve is to drive the state of $\mathcal{P}_i$ from its current value to its value in $\mathcal{I}_{i+1}$. Let us note Γ as a path from $\mathcal{F}_i$ to $\mathcal{F}_{i+1}$. The control strategy consists of guiding $\mathcal{I}_c$ to $\mathcal{I}_{i+1}$ by asymptotically regulating the axle $\mathbf{Y}_c$ on Γ. The control objective is achieved if $\mathbf{Y}_c$ is regulated to Γ before the origin of $\mathcal{F}_c$ reaches the origin of $\mathcal{F}_{i+1}$.

2) Vehicle Modeling: Our experimental vehicle is devoted to urban transportation, i.e., it moves on asphalt-even grounds at rather slow speeds. Therefore, it appears quite natural to rely on a kinematic model and to assume pure rolling and nonslipping

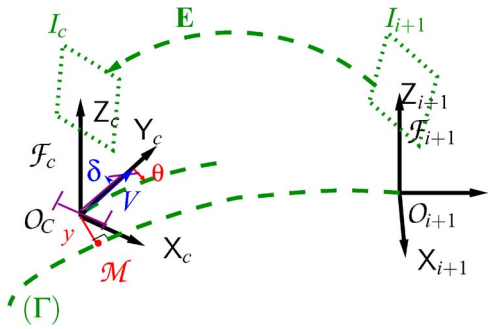


Fig. 5. Images $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ are two consecutive key images of the visual route Ψ. $\mathcal{I}_c$ is the current image. Γ is the path to follow.

at the wheel–ground contact. In such cases, vehicle modeling is commonly achieved, for instance, by relying on Ackermann's model, which is also named the bicycle model: the two front wheels located at the mid-distance between actual front wheels and actual rear wheels. As previously seen, our control problem has as an objective that the vehicle follows a reference path Γ. We propose to describe here its configuration with respect to that path rather than with respect to an absolute frame. To meet this objective, the following notations are introduced (see Fig. 5).

1) O_C is the center of the vehicle rear axle.
2) $\mathcal{M}$ is the point of Γ that is closest to O_C. This point is assumed to be unique, which is realistic when the vehicle remains close to Γ.
3) s is the curvilinear coordinate of point $\mathcal{M}$ along Γ, and $c(s)$ denotes the curvature of Γ at that point.
4) y and θ are, respectively, the lateral and angular deviations of the vehicle with respect to the reference path Γ.
5) δ is the virtual front wheel steering angle.
6) V is the linear velocity along the axle $\mathbf{Y}_c$ of $\mathcal{F}_c$.
7) l is the vehicle wheelbase.

The vehicle configuration can be described without ambiguity by the state vector (s, y, θ): the first two variables provide point O_C location, and the last variable denotes the vehicle heading. Since V is considered a parameter, the only control variable available to achieve path following is δ. The vehicle kinematic model can then be derived by writing that the velocity vectors at point O_C and at the center of the front wheel are directed along the wheel planes and that the vehicle motion is, at each instant, a rotation around an instantaneous rotation center. Such calculations lead to (see [19])

$$\begin{cases} \dot{s} = V \frac{\cos\theta}{1 - c(s)y} \\ \dot{y} = V \sin\theta \\ \dot{\theta} = V \left(\frac{\tan\delta}{l} - \frac{c(s)\cos\theta}{1 - c(s)y} \right). \end{cases} \tag{1}$$

Model (1) is clearly singular when $y = (1/c(s))$, i.e., when point O_C is superposed with the path Γ curvature center at abscissa s. However, this configuration is never encountered in practical situations: on one hand, the path curvature is small, and on the other hand, the vehicle is expected to remain close to Γ.

B. Control Design

The control objective is to ensure the convergence of y and θ toward 0 before the origin of $\mathcal{F}_c$ reaches the origin of $\mathcal{F}_{i+1}$. The vehicle model [see (1)] is clearly nonlinear. However, it has been established in [20] that mobile robot models can generally be converted in an exact way into almost linear models named chained forms. This property offers two very attractive features: on one hand, the path following control law can be designed and tuned according to celebrated Linear System Theory while nevertheless controlling the actual nonlinear vehicle model. Control law convergence and performances are then guaranteed whatever the vehicle initial configuration is. On the other hand, the chained form enables to specify, in a very natural way, the control law in terms of the distance covered by the vehicle rather than time. Vehicle spatial trajectories can then easily be controlled whatever the vehicle velocity is [21].

The conversion of the vehicle model [(1)] into chained form can be achieved due to the following state and control transformation:

$$\begin{aligned}\Phi\left([s \quad y \quad \theta]\right) &= [a_1 \quad a_2 \quad a_3] \\ &\triangleq [s \quad y \quad (1-c(s)y)\tan(\theta)] \qquad (2)\end{aligned}$$

$$(m_1, m_2) = \Psi(V, \delta) \qquad (3)$$

with

$$m_1 \triangleq V\frac{\cos\theta}{1-c(s)y} \qquad (4)$$

$$m_2 \triangleq \frac{d}{dt}\left((1-c(s)y)\tan(\theta)\right). \qquad (5)$$

Substituting (2), (4), and (5) into (1) establishes that the nonlinear model [(1)] can be rewritten, without approximation, as the standard chained form

$$\begin{cases}\dot{a}_1 = m_1 \\ \dot{a}_2 = a_3\, m_1 \\ \dot{a}_3 = m_2.\end{cases} \qquad (6)$$

To verify that a chained system is almost linear, replace the time derivative by a derivation with respect to the state variable a_1. Using the notations

$$\frac{da_i}{da_1} = a_i'$$

$$m_3 = \frac{m_2}{m_1}$$

the chained form [(6)] can be rewritten as

$$\begin{cases}a_1' = 1 \\ a_2' = a_3 \\ a_3' = m_3.\end{cases} \qquad (7)$$

The last two equations of (7) clearly constitute a linear system. Path following can now be easily addressed: in view of (2), the desired convergence of $(y\theta)$ to 0 is equivalent to those of $(a_2 a_3)$. The convergence of these two latter variables can easily be achieved by designing the auxiliary control input as

$$m_3 = -K_d a_3 - K_p a_2 \quad (K_p, K_d) \in \mathcal{R}^2 \qquad (8)$$

since reporting (8) in (7) provides

$$\frac{da_2'}{da_1} + K_d a_2' + K_p a_2 = 0. \qquad (9)$$

Moreover, since the evolution of the error dynamics [(9)] is driven by $a_1 = s$ (distance covered by the vehicle along reference path Γ), the gains (K_d, K_p) impose a settling distance instead of a settling time. Consequently, for a given initial error, the vehicle trajectory will be identical whatever the value of V is and even if V is time varying $(V \neq 0)$. The control law performances are, therefore, velocity independent. The study of the second-order differential equation [see (9)] can allow us to fix the gains (K_d, K_p) for desired control performances.

The expression of the actual control law δ can finally be obtained by inverting the chained transformation as

$$\begin{aligned}\delta(y,\theta) = \arctan &\times \Bigg(-l\Bigg[\frac{\cos^3\theta}{(1-c(s)y)^2} \\ &\times \Bigg(\frac{dc(s)}{ds} y\tan\theta - K_d\,(1-c(s)y)\tan\theta \\ &\quad - K_p y + c(s)\,(1-c(s)y)\tan^2\theta\Bigg) \\ &+ \frac{c(s)\cos\theta}{1-c(s)y}\Bigg]\Bigg). \qquad (10)\end{aligned}$$

In our experiments, the path to follow is simply defined as the straight line $\Gamma' = (O_{i+1}, \mathbf{Y}_{i+1})$ (see Fig. 5). In this case, $c(s) = 0$, and the control law [see (10)] can be simplified as

$$\delta(y,\theta) = \arctan\left(-l\left[\cos^3\theta(-K_d\tan\theta - K_p y)\right]\right). \qquad (11)$$

The implementation of the control law [(11)] requires the online estimation of the lateral deviation y and the angular deviation θ of $\mathcal{F}_c$ with respect to Γ. In the next section, we describe how the geometrical relationships between two views acquired with a camera under the generic projection model (conventional, catadioptric, and fisheye cameras) are exploited to enable a partial Euclidean reconstruction from which (y, θ) are derived.

C. State Estimation From the Generic Camera Model

Conventional cameras suffer from a restricted field of view. Many applications in vision-based robotics, such as the one proposed in this paper, can benefit from the panoramic field

of view provided by omnidirectional cameras. In practice, it is highly desirable that such imaging systems have a single viewpoint [7], [22]. That is, there exists a single center of projection so that every pixel in the sensed images measures the irradiance of the light passing through the same viewpoint in one particular direction. In this paper, we propose to use the unified model described in [23] since it allows to formulate state estimations that are valid for any sensor that obeys the unified camera model. In other words, it encompasses all the sensors in this class [23]: perspective and catadioptric cameras. A large class of fisheye cameras are also concerned by this model [8], [9], [24].

1) Camera Model: The unified projection model consists of a central projection onto a virtual unitary sphere followed by a perspective projection onto the image plane [23]. This generic model is parameterized by ξ describing the type of sensor and by a matrix $\mathbf{K}$ containing the intrinsic parameters. The coordinates $\mathbf{x}_i$ of the point in the image plane corresponding to the 3-D point $\mathcal{X}$ are obtained after three steps.

Step 1: The world points $\mathcal{X}$ of coordinates $\mathbf{X} = [XYZ]^T$ in the camera frame $\mathcal{F}_m$ are projected onto the unit sphere on a point $\mathcal{X}_m$ of coordinates $\mathbf{X}_m$ in $\mathcal{F}_m$: $\mathbf{X_m} = \mathbf{X}/\mathbf{X}\|\mathbf{X}\|$.

Step 2: The point coordinates are then changed to a new reference frame $\mathcal{F}_c$ centered in $C = (0, 0, -\xi)$ and perspectively projected onto the normalized image plane $Z = 1 - \xi$ as

$$\underline{\mathbf{x}}^\top = [\mathbf{x}^\top \; 1] = [x, y \; 1] = f(\mathbf{X}) = \left[\frac{X}{Z + \xi\|\mathbf{X}\|} \quad \frac{Y}{Z + \xi\|\mathbf{X}\|} \quad 1\right]. \tag{12}$$

Step 3: Finally, the coordinates $\underline{\mathbf{x}}_i^\top = [\mathbf{x}_i^\top 1]$ in the image plane are obtained after a plane-to-plane colineation $\mathbf{K}$ of the 2-D projective point $\underline{\mathbf{x}}$: $\underline{\mathbf{x}}_i = \mathbf{K}\mathbf{x}$.

We highlight that $\mathbf{X}_m$ can be computed as a function of the coordinates in the image and the sensor parameter ξ as

$$\mathbf{X}_m = (\eta^{-1} + \xi)\overline{\mathbf{x}}$$
$$\overline{\mathbf{x}} = \begin{bmatrix} \mathbf{x}^T & \frac{1}{1+\xi\eta} \end{bmatrix}^T \tag{13}$$

with

$$\begin{cases} \eta = \frac{-\gamma - \xi(x^2+y^2)}{\xi^2(x^2+y^2)-1} \\ \gamma = \sqrt{1 + (1-\xi^2)(x^2+y^2)}. \end{cases}$$

2) Scaled Euclidean Reconstruction: Let $\mathcal{X}$ be a 3-D point with coordinates $\mathbf{X_c} = [X_c Y_c Z_c]^T$ in the current frame $\mathcal{F}_c$ and $\mathbf{X}^* = [X_{i+1} Y_{i+1} Z_{i+1}]^T$ in the frame $\mathcal{F}_{i+1}$. Let $\mathbf{X_m}$ and $\mathbf{X}_m^*$ be the coordinates of those points projected onto the unit sphere (see Fig. 6). The epipolar plane contains the projection centers O_c and O_{i+1} and the 3-D point $\mathbf{X}$. $\mathbf{X_m}$ and $\mathbf{X}_m^*$ clearly belong to this plane. The coplanarity of those points leads to the relation

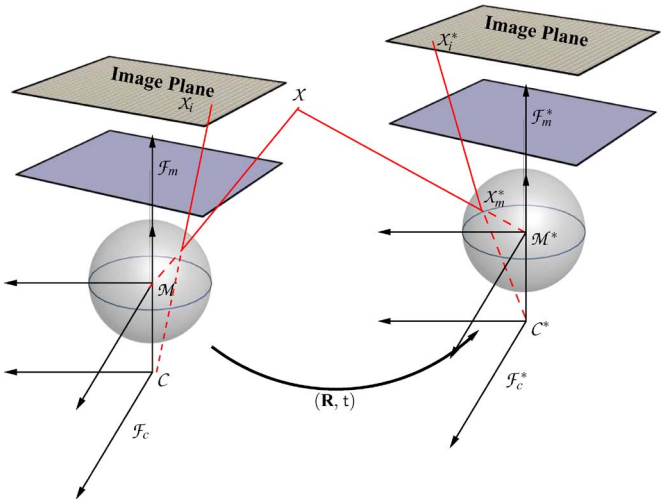


Fig. 6. Geometry of two views.

$$\mathbf{X_m^T R}\left(\mathbf{t} \times \mathbf{X_m^{*T}}\right) = \mathbf{X_m^T R}[\mathbf{t}]_\times \mathbf{X_m^{*T}} = 0 \tag{14}$$

where $\mathbf{R}$ and $\mathbf{t}$ represent the rotational matrix and the translational vector between the current and the desired frames. As in the case of the pinhole model, (14) may be written as

$$\mathbf{X_m^T E X_m^{*T}} = 0 \tag{15}$$

where $\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times$ is the essential matrix [22].

In (15), $\mathbf{X_m}$ (respectively, $\mathbf{X_m}^*$) corresponds to the coordinates of the point projected onto the sphere in the current image $\mathcal{I}_c$ (respectively, in the desired key image). Those coordinates are obtained thanks to the (13) and to the coordinates of the point matched in the first and second images. The essential matrix $\mathbf{E}$ between two images can be estimated by using five couples of matched points as proposed in [25] if the camera calibration (matrix $\mathbf{K}$) is known. Outliers are rejected using a random sample consensus (RANSAC) algorithm [26]. From the essential matrix, the camera motion parameters (that is, the rotation $\mathbf{R}$ and the translation $\mathbf{t}$ up to a scale) can be determined (see [27]). Finally, the estimation of the input of the control law [see (11)], i.e., the angular deviation θ and the lateral deviation y, can straightforwardly be computed from $\mathbf{R}$ and $\mathbf{t}$.

V. Experimentations

A. Experimental Setup

Our experimental vehicle is depicted in Fig. 7. It is an urban electric vehicle, which is named RobuCab, manufactured by Robosoft Company. Currently, RobuCab serves as the experimental testbed in several French laboratories. The four dc motors are powered by lead-acid batteries, providing 2 h of

Fig. 7. RobuCab vehicle with the embedded camera.

Fig. 8. Some key images of the memory.

autonomy. Vision and guidance algorithms are implemented in C^{++} language on a laptop using RTAI-Linux OS with a 2-GHz Centrino processor. The Fujinon fisheye lens mounted onto a Marlin F131B camera has a field-of-view of 185°. The image resolution in the experiments was 800 × 600 pixels. It has been calibrated using the Matlab toolbox presented in [24]. The camera, looking forward, is situated at approximately 80 cm from the ground. The parameters of the rigid transformation between the camera and the robot control frames are roughly estimated. Grey-level images are acquired at a rate of 15 ft/s.

B. Learning Step

In our experiment, RobuCab is manually driven along the 800-m-long path drawn in blue in Fig. 10. This path contains important turns as well as ways down and up and a come back. After the selection step, 800 key images are kept and form the visual memory of the vehicle. Some of those images are represented in Fig. 8.

C. Localization Step and Initialization

The navigation task has been started near the visual route to follow [the corresponding image is shown in Fig. 9(a)]. In this configuration, 15 images of the visual memory have been used in the first stage of the localization process. The

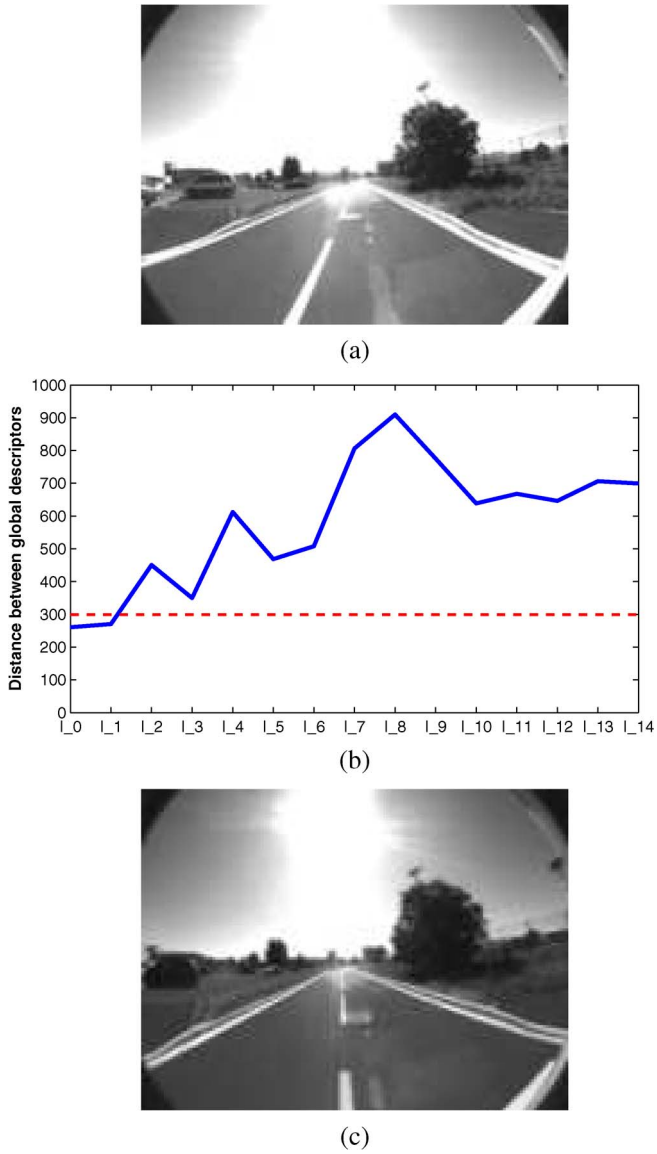


Fig. 9. Localization step. $\mathcal{I}_s$ is the current initial image. The distance between the current initial image and the key image global descriptors is drawn in (b). After using the local descriptors, $\mathcal{I}_s^*$ is selected as the correct image.

distances between the global descriptor of the current image and the descriptor of the memorized images (computed offline) are obtained as presented in Section III-A [see Fig. 9(b)]. After the second step of the localization process, the image shown in Fig. 9(c) is chosen as the closest to the image in Fig. 9(a). Given a goal image, a visual route starting from $\mathcal{I}_i^*$ and composed of 750 key images has been extracted from the visual memory.

D. Autonomous Navigation

The control [see (11)] is used to drive the vehicle along the visual route. A key image is assumed to be reached if the "image error" is smaller than a fixed threshold. In our experiment, the "image error" has been defined as the longest distance (expressed in pixels) between an image point and its position in the desired key image. The longitudinal velocity V

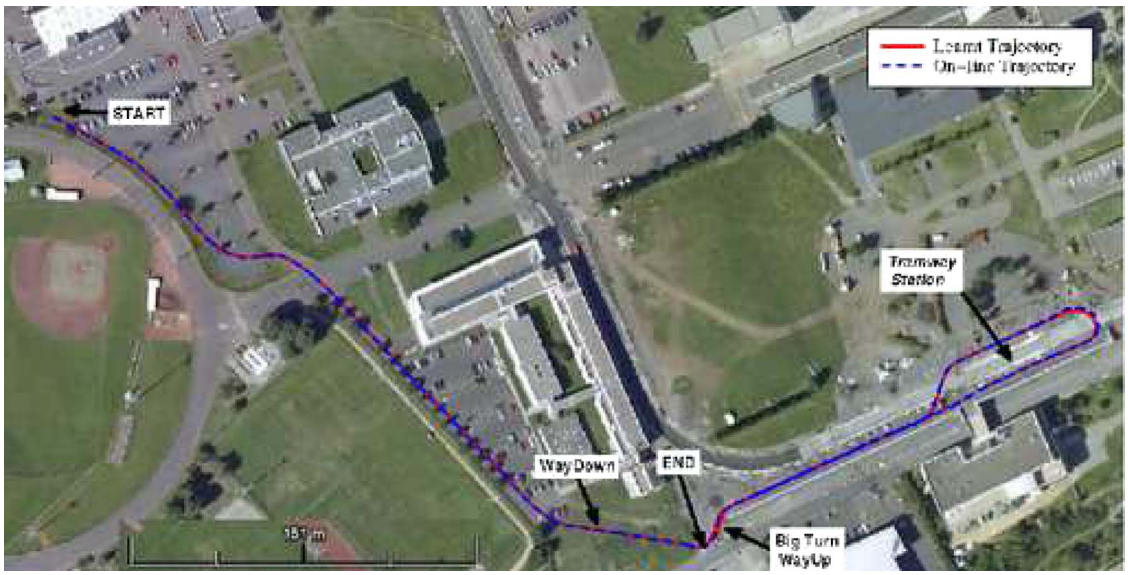


Fig. 10. Paths in the universitary campus executed during the memorization step (in red) and the autonomous step (in blue).

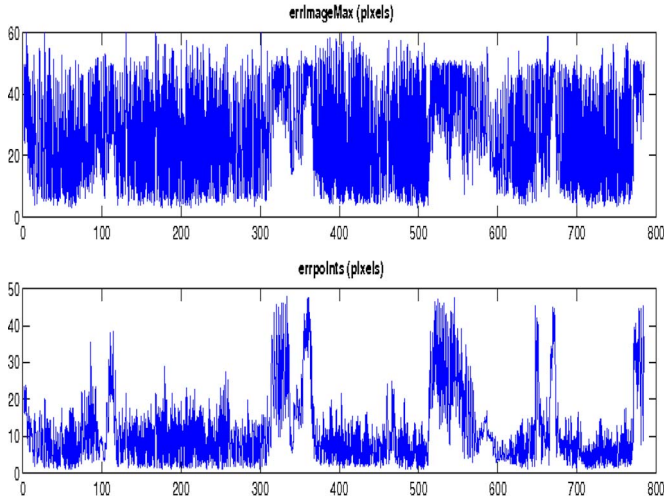


Fig. 11. Errors in the images (in pixels) versus time (in seconds).

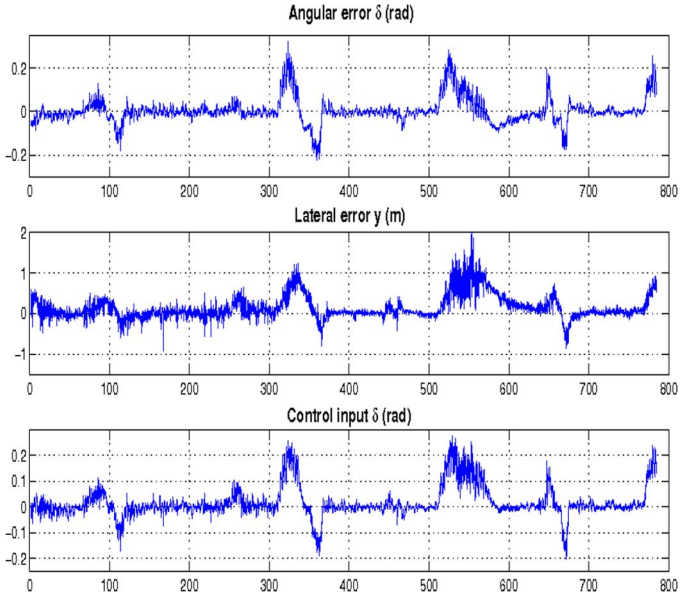


Fig. 12. Lateral y and angular θ errors, and control input δ versus time.

is fixed between 1 and 0.4 ms^{-1}. K_p and K_d have been set so that the error presents a double pole located at a value of 0.3. The vehicle successfully follows the learned path (see Fig. 10). The experiment lasts 13 min for a path of 754 m. A mean of 123 robust matches for each frame has been found. The mean computational time during the online navigation was 82 ms by image. As can be observed in Fig. 11, the errors in the images decrease to zero until reaching a key image. The lateral and angular errors as well as the control inputs are represented in Fig. 12. As can be seen, those errors are well regulated to zero for each key view. Discontinuities due to transitions between two successive key images can also be observed in Fig. 12.

Some of the reached images (with the corresponding images of the memory) are shown in Fig. 13. Note that illumination conditions have changed between the memorization and the autonomous steps (see Fig. 13(a) and (b), for example) as well as the contents (see Fig. 13(i) and (j), where a tram masks many visual features during the autonomous navigation).

Evaluation With a RTKGPS: The experimental vehicle has been equipped with a real-time kinematic differential GPS (RTKGPS; Thales Sagitta model). It is accurate to 1 cm (standard deviation) in the horizontal plane when enough satellites are available. The accuracy on a vertical axis is only 20 cm on our hardware platform. We thus discard the vertical readings, and the reported errors are measured in the horizontal plane.

Differential GPS (DGPS) data have been recorded during the learning and autonomous stages. The results are reported in Fig. 14. The red and blue plain lines represent, respectively, the trajectories recorded during the learning and autonomous stages. It can be observed that these trajectories are similar.

The distances (lateral error) between the vehicle positions during the learning and autonomous stages are reported in Fig. 14. The mean of the lateral error is about 25 cm with a standard deviation of 34 cm. The median error is less than

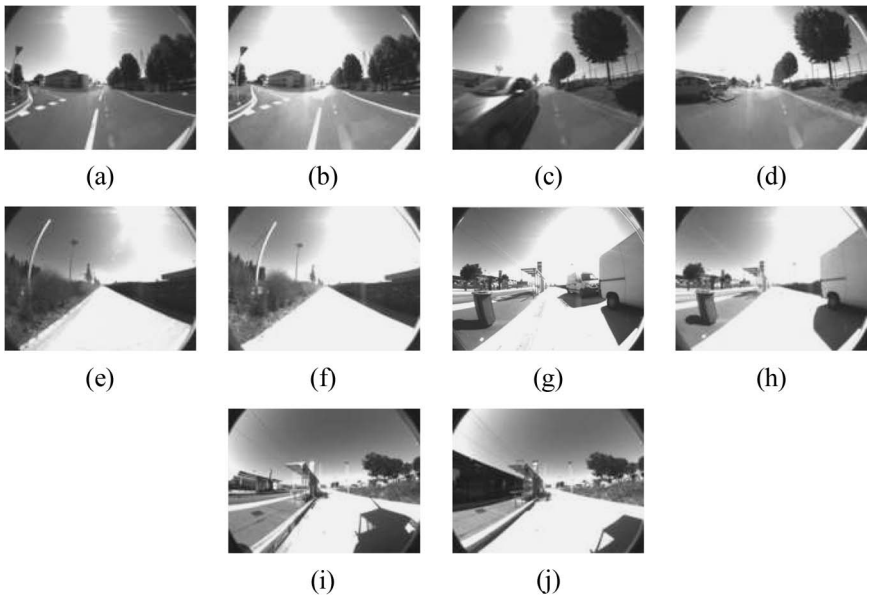


Fig. 13. Some of the current images $\mathcal{I}_k^r$ where the key images $\mathcal{I}_k$ have been reached. (a) $\mathcal{I}_a$. (b) $\mathcal{I}_a^r$. (c) $\mathcal{I}_b$. (d) $\mathcal{I}_b^r$. (e) $\mathcal{I}_c$. (f) $\mathcal{I}_c^r$. (g) $\mathcal{I}_d$. (h) $\mathcal{I}_d^r$. (i) $\mathcal{I}_e$. (j) $\mathcal{I}_e^r$.

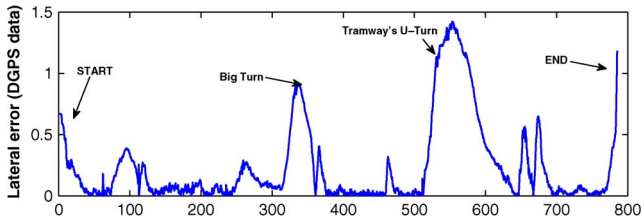


Fig. 14. Lateral error (distance between the autonomous and the learned trajectories, expressed in meters) obtained from DGPS data versus time.

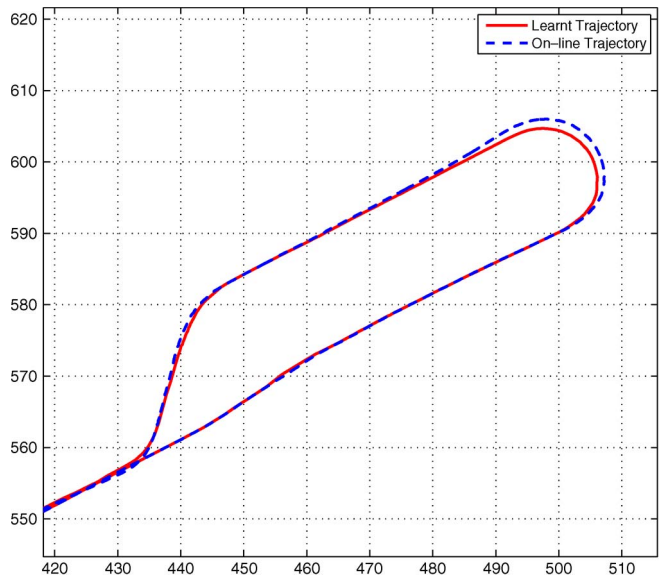


Fig. 15. Zoom on the trajectories around the tramway station (positions are expressed in meters).

10 cm. The maximal errors are observed along severe turns (see Fig. 15, representing a U-turn nearby the tramway station). Note that despite those errors, the visual path is still satisfactorily executed (after some images, the vehicle is still at a small distance from the learned trajectory).

VI. Conclusion

We have presented a complete framework for autonomous navigation that enables a vehicle to follow a visual path obtained during a learning stage using a single camera and natural landmarks. The robot environment is modeled as a graph of visual paths called visual memory from which a visual route connecting the initial and goal images can be extracted. The robotic vehicle can then be driven along the visual route thanks to a vision-based control law that takes into account nonholonomic constraints. Furthermore, the state of the robot is estimated using a generic camera model that is valid for a perspective, catadioptric, and large class of fisheye cameras. Our approach has been validated on an urban vehicle navigating along a 750-m-long trajectory. The experiments have shown that the navigation strategy is robust to some changes between the learning and autonomous navigation steps.

References

[1] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 237–267, Feb. 2002.

[2] J. Hayet, F. Lerasle, and M. Devy, "A visual landmark framework for indoor mobile robot navigation," in *Proc. IEEE ICRA*, Washington, DC, May 2002, pp. 3942–3947.

[3] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *Int. J. Comput. Vis.—Special Joint Issue on Vision and Robotics*, vol. 74, no. 3, pp. 237–260, Sep. 2007.

[4] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," in *Proc. IEEE ICRA*, Minneapolis, MN, Apr. 1996, vol. 1, pp. 83–88.

[5] S. Jones, C. Andresen, and J. Crowley, "Appearance-based process for visual navigation," in *Proc. IEEE/RSJ Int. Conf. IROS*, Grenoble, France, 1997, vol. 2, pp. 551–557.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Conf.*, 1988, pp. 189–192.

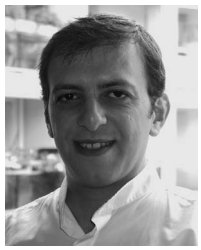
[7] S. Baker and S. K. Nayar, "A theory of single-viewpoint catadioptric image formation," *Int. J. Comput. Vis.*, vol. 35, no. 2, pp. 1–22, Nov. 1999.

[8] J. Barreto, "A unifying geometric representation for central projection systems," *Comput. Vis. Image Underst.—Special Issue on Omnidirectional Vision and Camera Networks*, vol. 103, no. 3, pp. 208–217, Sep. 2006.

[9] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet, "A generic fisheye camera model for robotic applications," in *Proc. IEEE/RSJ Int. Conf. IROS*, San Diego, CA, Oct. 29–Nov. 2, 2007, pp. 1683–1688.
[10] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue, "Visual navigation using omnidirectional view sequence," in *Proc. IEEE/RSJ Int. Conf. IROS*, Kyongju, Korea, Oct. 1999, vol. 1, pp. 317–322.
[11] T. Pajdla and V. Hlaváè, "Zero phase representation of panoramic images for image based localization," in *Proc. 8th Int. Conf. Comput. Anal. Images Patterns*, F. Solina and A. Leonardis, Eds. Berlin, Germany: Springer-Verlag, 1999, vol. 1689, pp. 550–557.
[12] J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based navigation and environmental representations with an omnidirectional camera," *IEEE Trans. Robot. Autom.*, vol. 16, no. 6, pp. 890–898, Dec. 2000. VisLab-TR 12/2000.
[13] T. Goedemé, T. Tuytelaars, G. Vanacker, M. Nuttin, L. V. Gool, and L. V. Gool, "Feature based omnidirectional sparse visual path following," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 1806–1811.
[14] A. Tamimi, H. Andreasson, A. Treptow, T. Duckett, and A. Zell, "Localization of mobile robots with omnidirectional vision using particle filter and iterative SIFT," in *Proc. ECMR*, 2005, pp. 1–7.
[15] A. Murillo, C. Sagüés, J. Guerrero, T. Goedemé, T. Tuytelaars, and L. V. Gool, "From omnidirectional images to hierarchical localization," *Robot. Auton. Syst.*, vol. 55, no. 5, pp. 372–382, May 2007.
[16] E. Menegatti, T. Maeda, and H. Ishiguro, "Image-based memory for robot navigation using properties of omnidirectional images," *Robot. Auton. Syst.*, vol. 47, no. 4, pp. 251–267, Jul. 2004.
[17] P.-O. Persson and G. Strang, "A simple mesh generator in MATLAB," *SIAM Rev.*, vol. 46, no. 2, pp. 329–345, Jun. 2004.
[18] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet, "Efficient hierarchical localization method in an omnidirectional images memory," in *Proc. IEEE ICRA*, Pasadena, CA, May 19–23, 2008, pp. 13–18.
[19] T. Zodiac, *Theory of Robot Control*, C. Canedas de Wit, B. Siciliano, and G. Bastin, Eds. Berlin, Germany: Springer-Verlag, 1995.
[20] C. Samson, "Control of chained systems application to path following and time-varying stabilization of mobile robots," *IEEE Trans. Autom. Control*, vol. 40, no. 1, pp. 64–77, Jan. 1995.
[21] B. Thuilot, J. Bom, F. Marmoiton, and P. Martinet, "Accurate automatic guidance of an urban electric vehicle relying on a kinematic GPS sensor," in *Proc. 5th IFAC Symp. IAV*, Jul. 5–7, 2004.
[22] T. Svoboda and T. Pajdla, "Epipolar geometry for central catadioptric cameras," *Int. J. Comput. Vis.*, vol. 49, no. 1, pp. 23–37, Aug. 2002.
[23] C. Geyer and K. Daniilidis, "Mirrors in motion: Epipolar geometry and motion estimation," *Int. J. Comput. Vis.*, vol. 45, no. 3, pp. 766–773, 2003.
[24] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 3945–3950.
[25] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, Jun. 2004.
[26] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
[27] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

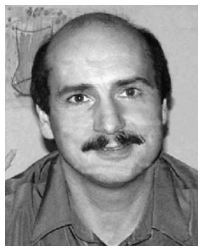
Jonathan Courbon received the degree from Institut Franćais de Mécanique Avancée (IFMA), Aubiere, France, in 2005. He is currently working toward the Ph.D. degree with the Robotics and Vision Group, Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique–Centre National de la Recherche Scientifique, Aubiere.

He is also with CEA, List, Fontenay Aux Roses, France. His research focuses on the use of visual information for the navigation and control of mobile robots.



Youcef Mezouar received the Ph.D. degree in computer science from the University of Rennes 1, Rennes, France, in 2001.

He was a Postdoctoral Associate with the Robotics Laboratory, Computer Science Department, Columbia University, New York, NY. He is currently an Associate Professor with the Blaise Pascal University, Clermont-Ferrand, France, and with Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique, Aubiere, France. His research interests include robotics, microrobotics, computer vision, and vision-based control and navigation.



Philippe Martinet received the degree from Centre Universitaire des Sciences et Technologies (CUST), Clermont-Ferrand, France, in 1985 and the Ph.D. degree in electronics science from the Blaise Pascal University, Clermont-Ferrand, in 1987.

From 1990 to 2000, he was an Assistant Professor with the Electrical Engineering Department, CUST. Since 2000, he has been a Professor with the Institut Franćais de Mécanique Avancée (IFMA), Aubiere, France. He was the leader of the group Groupe de Recherche en Automatique VIsion et Robotique (over 74 people) from 2001 to 2006. He is performing research at the Robotics and Vision Group, LASMEA-CNRS, Aubiere. His research interests include visual servoing, force-vision coupling, multisensor-based control, autonomous vehicles, and modeling and control of complex machines.