

A generic framework for topological navigation of urban vehicle

Jonathan Courbon, Youcef Mezouar, Laurent Eck, Philippe Martinet

► **To cite this version:**

Jonathan Courbon, Youcef Mezouar, Laurent Eck, Philippe Martinet. A generic framework for topological navigation of urban vehicle. ICRA09 - Workshop on Safe navigation in open and dynamic environments Application to autonomous vehicles, May 2009, Kobe, Japan. hal-02468609

HAL Id: hal-02468609

<https://hal.inria.fr/hal-02468609>

Submitted on 6 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A generic framework for topological navigation of urban vehicle

Jonathan Courbon^{*†}, Youcef Mezouar[†], Laurent Eck^{*}, Philippe Martinet[†]

^{*}CEA, List

[†]LASMEA

18 route du Panorama, BP6

24 Avenue des Landais

F- 92265 FONTENAY AUX ROSES - FRANCE

63177 AUBIERE - FRANCE

Email: laurent.eck@cea.fr

Email: firstname.lastname@lasmea.univ-bpclermont.fr

Abstract—In this paper, we present a generic framework for urban vehicle navigation using a topological map. This map is built by taking into account the non-holonomic behaviour of the vehicle. After a localization step, a sensory route is extracted to reach a goal. This route is followed using a sensor-based control strategy, based on the vehicle model and computed from the state extracted from the current and the desired sensory images. In that aim, a generic model is proposed for visual sensors. Experiments with an urban electric vehicle navigating in an outdoor environment have been carried out with a fisheye camera using a single camera and natural landmarks. A navigation along a 1700-meter-long trajectory validates our approach.

Index Terms—Urban vehicle navigation, topological map, generic camera model, autonomous navigation, non-holonomic mobile vehicle, real-time application

I. INTRODUCTION

Saturation of vehicles traffic in large cities is a major concern. Improvements can be gained from the development of alternative public transportation systems. In order to meet public expectation, such systems should be very flexible, in order to be suitable answer to many different individual needs, and as nuisance free as possible (with respect to pollution, noise, urban scenery, ...). Individual vehicles, available in a car-sharing concept, meet clearly both requirements. They appear to be very suitable in specific areas where the public demand is properly structured, as in airport terminals, attraction resorts, university campus, or inner-cities pedestrian zones. In order to spread such a transportation system, 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 be brought back autonomously to stations for refilling and reuse. Automatic navigation is generally divided in four steps : 1) map building, 2) localisation onto the map, 3) path planning and 4) control to actually achieve the navigation task. Many works deal with the problems of fuzzing steps 1) and 2) on a single stage (Simultaneous Localization And Mapping; SLAM). Unfortunately, even if computers are more and more powerfull, those strategies are restricted to small environments since the computational cost highly increases with the number of features integrated onto the map. An alternative solution, suitable for large scale environment, consists on using a Geographical Information System (GIS) as proposed in [1].

Using visual sensors, appearance-based or “visual memory-based” navigation approaches are emerging. The main idea is to represent the mobile robot environment with a bounded quantity of images gathered in a database (visual memory). For example, [2] proposes to use a sequence of images recorded during a human teleoperated motion, and called View-Sequenced Route Reference. Such a strategy is called “mapless” (refer to [3]). Indeed, any notion of map nor topology of the environment appears, neither to build the reference set of images, nor for the automatic guidance of the mobile robot. Similar approaches have been proposed for urban vehicles in [4]. In practise, a topological organization decreases the computational cost and is more intuitive.

In this paper, we present a generic framework for urban vehicle navigation using a topological map. This topological map directly takes into account the control constraints during its building (refer to Section II). Before the beginning of the motion the localization of the robotic system is performed. Given an image of one of the paths as a target, the vehicle navigation mission is defined as a concatenation of path subsets, called sensory route. A navigation task then consists in autonomously executing this route. The path-following control law adapted to the nonholonomic constraints of the vehicle is first defined (Section III-B). This control guides the vehicle along the reference route without explicitly planning any trajectory. This step requires also a model of the sensor to compute the state needed by the control law. In the case of visual sensors, we propose a generic model valid for a large set of cameras (including perspective, catadioptric, spherical and fisheye cameras). Those elements are presented in Section III-C. Experiments have been carried out with an electrical urban vehicle, navigating in outdoor environment along a 1200-meter-long trajectory. Results are presented in the last Section.

II. TOPOLOGICAL MAP WITH CONTROL CONSTRAINTS

In the sequel, we define an image I as the representation of the environment given by an embedded sensor. This sensor is supposed to be rigidly fixed to the vehicle. The environment is represented by a set of images, topologically organized.

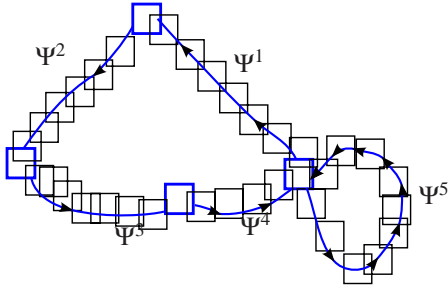


Fig. 1: The memory of the robot, composed of 5 ordered paths Ψ^i . Each square represents an image of the memory.

A. Representation of the environment

Let consider a sensory path Ψ^p composed of n key sensory images:

$$\Psi^p = \{I_i^p | i = \{1, 2, \dots, n\}\}$$

Such a path is a directed graph composed of images successively acquired. In practise, such a path can represent a street between two crossroads. This representation is justified because, when following the given path it is not necessary to take into account the other elements of the environment. Paths are then linked. The choice of the key images and the path linking is explained in Section II-C. The environment is thus represented by a topological map which is a multigraph of sensory paths (refer to Fig. 1).

B. Paths acquisition

The learning stage relies on the human experience. The user guides the vehicle along one or several paths of its workspace. During this stage, the motions are assumed to be limited to those of a car-like vehicle, which only goes forward. Images are acquired by the embedded sensor and then, a selection process occurs in order to keep only some images called “key images”. As noticed in [2], the number of key images of a visual path is directly linked to the human-guided path complexity.

C. Control constraints during map building

For control purpose (refer to Section III), the authorized motions are assumed to be limited to those of a car-like vehicle, which only goes forward. The following Hypothesis 2.1 formalizes these constraints.

Hypothesis 2.1: Given two frames ${}^R\mathcal{F}_i$ and ${}^R\mathcal{F}_{i+1}$, respectively associated to the vehicle when two successive key images I_i and I_{i+1} of a sensory 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 which only moves forward.

Moreover, because the controller is sensor-based, the robot is controllable from I_i to I_{i+1} only if the hereunder Hypothesis 2.2 is respected.

Hypothesis 2.2: Two successive key sensory images I_i and I_{i+1} contain a set \mathcal{P}_i of matched features, which can be tracked

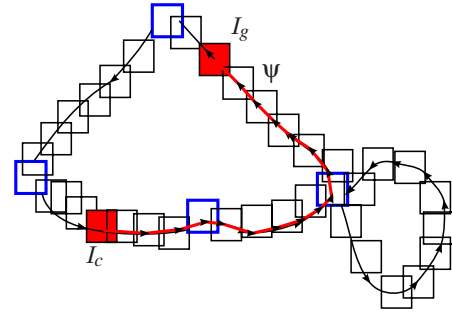


Fig. 2: A sensory route, from the current and goal locations (in the sensory memory) of the robot.

along a path performed between ${}^R\mathcal{F}_i$ and ${}^R\mathcal{F}_{i+1}$ and which are sufficient to compute the full control law.

This Hypothesis 2.2 has three effects. Firstly, it limits the set of possible sensors. In fact, some sensors may not provide enough information to compute the control law. Secondly, for the same reason, the position of the sensor is important as it must provide the needed information. Finally, during the acquisition of a sensory path, this hypothesis constrains the choice of the key images.

In order to connect two sensory paths, the terminal extremity of one of them and the initial extremity of the other one must be constrained as two consecutive key images of a sensory path.

D. Sensory route

A sensory route describes the vehicle’s mission in the sensor space. Given two key images of the sensory memory I_c and I_g , corresponding respectively to the current and goal locations of the robot, a sensory route ψ is a set of key images which describes a path from I_c to I_g (refer to Fig.2).

The sensory route describes a set of consecutive states that the sensor has to reach in order that the robot joins the goal configuration from the initial one. The robot motions are controlled along the sensory route using the data provided by the embedded sensor. In that aim, the sensor has to be modelled as well as the vehicle. A control law is then designed and computable by the state given by the sensor’s relative information. The next section deals with those issues.

III. MODELLING AND CONTROL

When starting the autonomous navigation task, the output of the localization step provides the closest image I_c to the current initial image I_c^* . A visual route Ψ connecting I_c to the goal image is then extracted from the visual memory. As previously explained, the sensory route is composed of a set of key images. The next step is to automatically follow this route using a sensor-based technique. The principle is presented in Fig. 3.

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

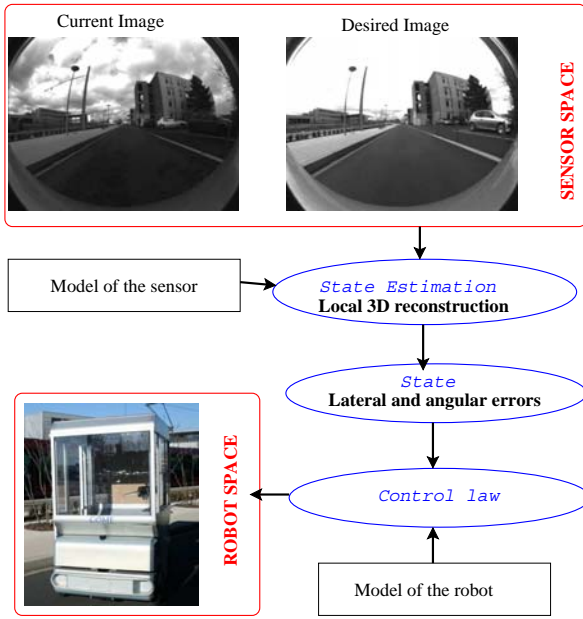


Fig. 3: Route following process with a visual sensor.

is formulated as a path following to guide the nonholonomic vehicle along the sensory route.

A. Model and assumptions

1) *Control objective*: Let I_i and I_{i+1} be two consecutive key images of a given route to follow and I_c be the current image. Let us note $\mathcal{F}_i = (O_i, \mathbf{X}_i, \mathbf{Y}_i, \mathbf{Z}_i)$ and $\mathcal{F}_{i+1} = (O_{i+1}, \mathbf{X}_{i+1}, \mathbf{Y}_{i+1}, \mathbf{Z}_{i+1})$ the frames attached to the vehicle when I_i and I_{i+1} were stored and $\mathcal{F}_c = (O_c, \mathbf{X}_c, \mathbf{Y}_c, \mathbf{Z}_c)$ a frame attached to the vehicle in its current location. Figure 4 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 2.2, the state of a set of features \mathcal{P}_i is known in the images I_i and I_{i+1} . The state of \mathcal{P}_i is also assumed available in I_c . The task to achieve is to drive the state of \mathcal{P}_i from its current value to its value in I_{i+1} . Let us note Γ a path from \mathcal{F}_i to \mathcal{F}_{i+1} . The control strategy consists in guiding I_c to I_{i+1} by regulating asymptotically 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 Modelling*: 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 non slipping at wheel-ground contact. In such cases, the vehicle modelling is commonly achieved for instance relying on the Ackermann's model, also named the bicycle model: the two front wheels located at the mid-distance between actual front wheels and actual rear wheels. As seen previously, our control problem has as objective that the vehicle follows a reference path Γ , we propose to describe here, its configuration with

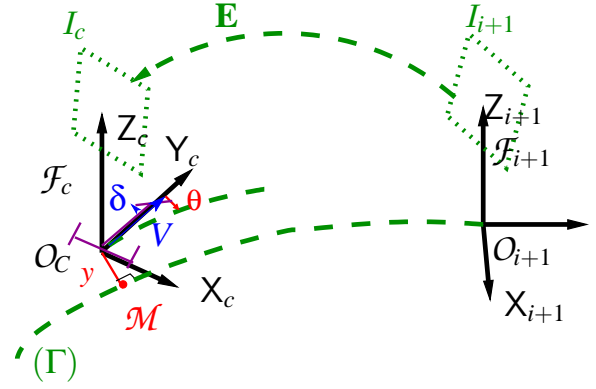


Fig. 4: Images I_i and I_{i+1} are two consecutive key images of the visual route Ψ . I_c is the current image. Γ is the path to follow.

respect to that path, rather than with respect to an absolute frame. To meet this objective, the following notations are introduced (see Figure 4).

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

Vehicle configuration can be described without ambiguity by the state vector (s, y, θ) : the two first variables provide point O_c location and the last one the vehicle heading. Since V is considered as a parameter, the only control variable available to achieve path following is δ . The vehicle kinematic model can then be derived by writing that velocity vectors at point O_c and at center of the front wheel are directed along wheel planes and that the vehicle motion is, at each instant, a rotation around an instantaneous rotation center. Such calculations lead to (refer to [5]):

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

Model (1) is clearly singular when $y = \frac{1}{c(s)}$ *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, 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 (1) is clearly nonlinear. However, it has been established in [6] 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, path following control law can be designed and tuned according to celebrated Linear System Theory, while controlling nevertheless the actual non linear vehicle model. Control law convergence and performances are then guaranteed whatever the vehicle initial configuration is. On the other hand, chained form enables to specify, in a very natural way, control law in term of distance covered by the vehicle, rather than in term of time. Vehicle spacial trajectories can then easily be controlled, whatever the vehicle velocity is [7].

Conversion of the vehicle model (1) into chained form can be achieved thanks to the following state and control transformation:

$$\Phi([s \ y \ \theta]) \triangleq [s \ y \ (1-c(s)y)\tan(\theta)] \quad (2)$$

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

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

The gains (K_d, K_p) impose a settling distance and set the desired control performances. 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 ($V \neq 0$). Control law performances are therefore velocity independent. In our experiments the path to follow is simply defined as the straight line $\Gamma' = (O_{i+1}, \mathbf{Y}_{i+1})$ (refer to Figure 4). In this case $c(s) = 0$ and the control law (3) can be simplified as follows:

$$\delta(y, \theta) = \arctan \left(-l \left[\cos^3 \theta (-K_d \tan \theta - K_p y) \right] \right) \quad (4)$$

The implementation of control law (4) requires the on-line 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 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 a visual sensor

Different sensors are suitable for our application. The method consists on two steps: 1/ sensor modeling, 2/ extraction of the state of the robot in the sensor space. In the sequel, visual cameras are used to extract the state required by the control law but our framework is not limited to those sensors. We consider a camera modeled by the generic projection on the sphere and the image of points features. The unified

projection model consists on a projection onto a virtual unitary sphere, followed by a perspective projection onto an image plane [8]. This virtual unitary sphere is centered in the principal effective view point and the image plane is attached to the perspective camera.

Let \mathcal{F}_c and \mathcal{F}_m be the frames attached to the conventional camera and to the unitary sphere respectively. In the sequel, we suppose that \mathcal{F}_c and \mathcal{F}_m are related by a simple translation along the Z-axis (\mathcal{F}_c and \mathcal{F}_m have the same orientation). The origins \mathcal{C} and \mathcal{M} of \mathcal{F}_c and \mathcal{F}_m will be termed optical center and principal projection center respectively. The optical center \mathcal{C} has coordinates $[0 \ 0 \ -\xi]^T$ with respect to \mathcal{F}_m and the image plane is orthogonal to the Z-axis and it is located at a distance $Z = f_c$ from \mathcal{C} .

Let X be a 3D point with coordinates $X = [X \ Y \ Z]^T$ with respect to \mathcal{F}_m . The point on the normalized image plane is of homogeneous coordinates $\underline{x} = [x^T \ 1]^T = f(X)$ (where $x = [x \ y]^T$):

$$\underline{x} = f(X) = \left[\begin{array}{ccc} X & Y & \\ \varepsilon_s Z + \xi \rho & \varepsilon_s Z + \xi \rho & 1 \end{array} \right]^T \quad (5)$$

The parameter ε_s allows to integrate the spherical projection into this model by setting $\varepsilon_s = 0$ and $\xi = 1$. In the general case and in the sequel, this parameter is equal to 1. Note that, setting $\xi = 0$ (and $\varepsilon_s = 1$), the general projection model becomes the well known pinhole model. ξ can be seen as a parameter which allows to control the amount of radial distortions for fisheye lenses.

Finally the point of homogeneous coordinates m in the image plane is obtained after a plane-to-plane collineation \mathbf{K} of the 2D projective point of coordinates \underline{x} :

$$m = \mathbf{K} \underline{x} \quad (6)$$

The matrix \mathbf{K} can be written as $\mathbf{K} = \mathbf{K}_p \mathbf{M}$ where the matrix \mathbf{K}_p contains the perspective camera intrinsic parameters, and the diagonal matrix \mathbf{M} links the frame attached to the unitary sphere to the camera frame \mathcal{F}_m . For a central catadioptric camera, this matrix depends on the shape of the mirror.

Let X be a 3D point with coordinates $X_c = [X_c \ Y_c \ Z_c]^T$ in the current frame \mathcal{F}_c and $X^* = [X_{i+1} \ Y_{i+1} \ Z_{i+1}]^T$ in the frame \mathcal{F}_{i+1} . Let X_m and X_m^* be the coordinates of those points, projected onto the unit sphere (refer to Fig. 5). Let \mathbf{R} (respectively \mathbf{t}) represent the rotational matrix (resp. the translational vector) between the current and the desired frames. Similarly to the case of pinhole model, the epipolar geometry leads to:

$$X_m^T \mathbf{E} X_m^* = 0 \quad (7)$$

where $\mathbf{E} = \mathbf{R}[\mathbf{t}]_{\times}$ is the essential matrix [9]. The essential matrix \mathbf{E} between two images is estimated using five couples of matched points as proposed in [10] if the camera calibration (matrix \mathbf{K}) is known. Outliers are rejected using a random sample consensus (RANSAC) algorithm. 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. Finally,

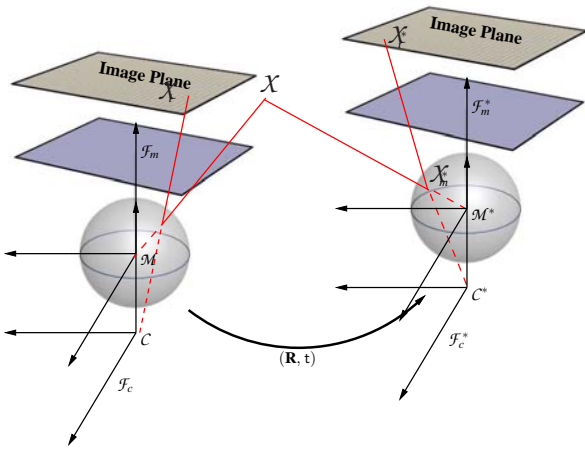


Fig. 5: Geometry of two views.

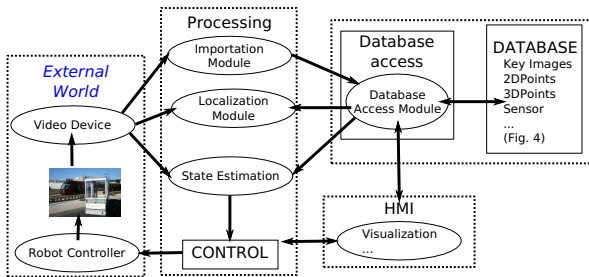


Fig. 6: Architecture of our software for visual navigation SoViN

the estimation of the input of the control law (3), *i.e.* the angular deviation θ and the lateral deviation y , are computed straightforwardly from \mathbf{R} and \mathbf{t} .

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

A. Map management

We have proposed a software platform called SoViN to efficiently manage visual memory for autonomous vehicle navigation in large scale environments [11]. An overview of SoViN is shown in Fig. 6. The software platform is divided into three different modules: a module for processing (image processing, computer vision and control); a module for HMI (visualization and high-level actions control) and a module for data storage and access (low-level functions). For data storage, our software uses a conventional database software. HMI and processing modules communicate with the database thanks to this low-level module.

During both localization and path following stages, key images' elements (image points with their descriptors, matching between successive image points ...) are loaded on-line from the database.

B. Experimental set-up

Our experimental vehicle is depicted in Figure 7. It is an urban electric vehicle, named RobuCab, manufactured by the



Fig. 7: RobuCab vehicle with the embedded camera.

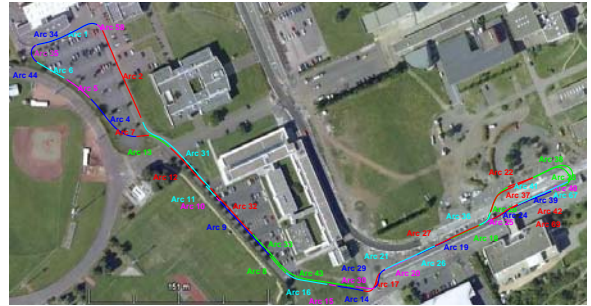


Fig. 8: Large-scale environment: large loop

Robosoft Company. Currently, RobuCab serves as experimental testbed in several French laboratories. The 4 DC motors are powered by lead-acid batteries, providing 2 hours autonomy. Vision and guidance algorithms are implemented in C++ language on a laptop using RTAI-Linux OS with a 2GHz Centrino processor. The Fujinon fisheye lens, mounted onto a Marlin F131B camera, has a field-of-view of 185 deg. The image resolution in the experiments was 800×600 pixels. It has been calibrated using the Matlab toolbox presented in [12]. The camera, looking forward, is situated at approximately 80cm 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 15fps.

C. Experimentations

1) *Learning stage:* The robot has been manually driven along a 1200 meter-long loop (refer to Fig. 8) at the beginning of July, with a very sunny weather. The fisheye lens camera was rigidly fixed at approximately 1 m from the ground, 1 m at the left of the middle of the car. The camera was looking in the direction of the vehicle. It has been calibrated using the unified model on the sphere. An importation step occurred and result to 35 edges after having cut some edges in function of the context (straight lines, huge turns). A longitudinal velocity has been given for each edge (0.4 m/s for huge turns, 0.8 m/s in small turns, 1 m/s for straight parts). The DGPS data has also been acquired. For each node, the position given by interpolation of the DGPS data have been saved too (interpolation in function of the time when data where acquired).

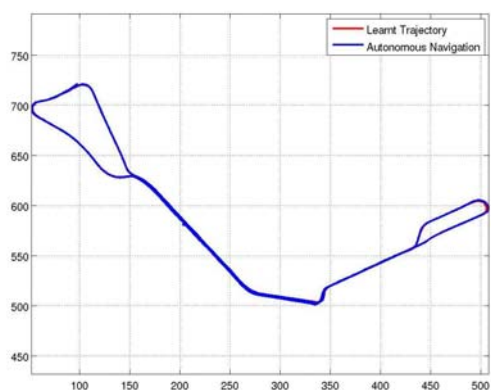


Fig. 11: Trajectories obtained with the RTK-DGPS

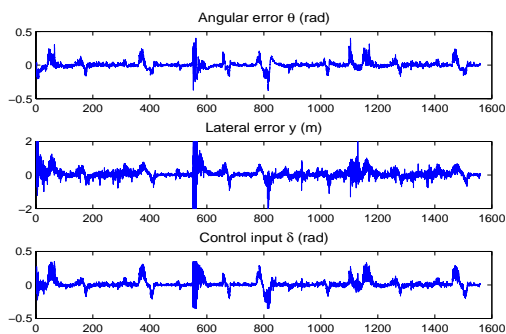


Fig. 9: Lateral y and angular θ errors and control input δ vs time (s).

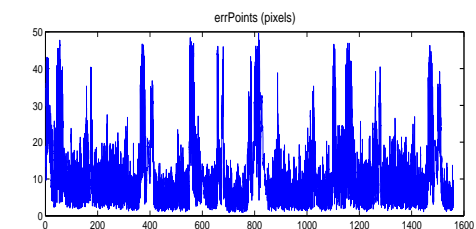


Fig. 10: Errors in the images (pixels) versus time (s).

2) *Localization step and initialisation*: After a localization step, a visual route is extracted. It consists on doing at least one loop.

3) *Autonomous navigation*: The experiment lasts 26 minutes for a path of 1700 meters (refer to Fig. 9) which results to an average longitudinal velocity of around 1 m/s. This visual route is composed of 54 edges and around 1400 key images. This path following stops for safety reasons because few visual features were robustly matched.

Evaluation with a RTK-GPS: DGPS data have been recorded during the learning and the autonomous stages. The results are reported in Fig. 11. The red and blue lines represents respectively the trajectories recorded during the learning and autonomous stages. It can be observed that these

trajectories are similar. The lateral error measured by the RTK-GPS has a mean of 23 cm and a standard deviation of around 30 cm.

V. CONCLUSION

We have presented a complete framework for autonomous navigation which enables a vehicle to follow a sensory path obtained during a learning stage. The robot environment is modeled as a topological map from which a sensory route connecting the initial and goal images can be extracted. The robotic vehicle can then be driven along the route thanks to a sensor based control law which takes into account non-holonomic constraints. Furthermore, the state of the robot is estimated using a generic camera model valid for a perspective, catadioptric as well as a large class of fisheye cameras. Our approach has been validated on an urban vehicle navigating along a long trajectory. At our knowledge, it reports it is the first time that a 1700-meter-long trajectory is done using a single camera and natural landmarks.

ACKNOWLEDGMENT

This work is supported by the EU-Project FP6 IST μ Drones, FP6-2005-IST-6-045248.

REFERENCES

- [1] P. Bonnifait, M. Jabbour, and V. Cherfaoui, "Autonomous Navigation in Urban Areas using GIS-Managed Information," *International Journal of Vehicle Autonomous Systems*, vol. 6, no. 1/2, pp. 83–103, 2008, special Issue on Advances in Autonomous Vehicles and Intelligent Transportation.
- [2] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," in *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 1, Minneapolis, Minnesota, April 1996, pp. 83–88.
- [3] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 237–267, february 2002.
- [4] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette, "Outdoor visual path following experiments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'07*, San Diego, CA, USA, 29 October - 2 November 2007, pp. 4265–4270.
- [5] T. Zodiak, *Theory of robot control*, C. C. de Wit, B. Siciliano, and G. Bastin, Eds. Springer-Verlag, Berlin, 1995.
- [6] C. Samson, "Control of chained systems. Application to path following and time-varying stabilization of mobile robots," *IEEE Transactions on Automatic Control*, vol. 40, no. 1, pp. 64–77, 1995.
- [7] B. Thuilot, J. Bom, F. Marmoiton, and P. Martinet, "Accurate automatic guidance of an urban electric vehicle relying on a kinematic GPS sensor," in *5th IFAC Symposium on Intelligent Autonomous Vehicles, IAV'04*, Instituto Superior Técnico, Lisbon, Portugal, July 5-7th 2004.
- [8] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," in *European Conference on Computer Vision*, vol. 29 (3), Dublin, Ireland, May 2000, pp. 159–179.
- [9] T. Svoboda and T. Pajdla, "Epipolar geometry for central catadioptric cameras," *International Journal of Computer Vision*, vol. 49, no. 1, pp. 23–37, 2002.
- [10] D. Nistér, "An efficient solution to the five-point relative pose problem," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.
- [11] J. Courbon, L. Lequievre, Y. Mezouar, and L. Eck, "Navigation of urban vehicle: An efficient visual memory management for large scale environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'08*, Nice, France, sep 2008.
- [12] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *IEEE International Conference on Robotics and Automation, ICRA'07*, Rome, Italy, apr 2007, pp. 3945–3950.