Autonomous navigation of vehicles from a visual memory using a generic camera model

Jonathan Courbon†, Youcef Mezouar*, Philippe Martinet*

*LASMEA
24 Avenue des Landais
63177 AUBIERE - FRANCE
Email: firstname.lastname@lasmea.univ-bpclermont.fr

†CEA, List
18 route du Panorama, BP6
F- 92265 FONTENAY AUX ROSES - FRANCE

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, a usual behavior consists of memorizing some key views along the performed path, in order to use these references as checkpoints for a future navigation mission. The navigation framework for wheeled vehicles presented in this paper is based on this assumption. During a human-guided learning step, the vehicle performs paths which are sampled and stored as a set of ordered key images, 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 route. When running autonomously, the control guides the vehicle along the reference visual route without explicitly planning any trajectory. The control consists of a vision-based control law adapted to the nonholonomic constraint. Our navigation framework has been designed for a generic class of cameras (including conventional, catadioptric and fish-eye cameras). Experiments with an urban electric vehicle navigating in an outdoor environment have been carried out with a fisheye camera along a 750-meter-long trajectory. Results validate our approach.

Index Terms—Robot navigation, monocular vision, visual memory, generic camera model, autonomous navigation, nonholonomic mobile vehicle, urban vehicles, 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 electric 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, universitary campus, or inner-cities pedestrian zones. In order to develop 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.

Such an automatic navigation may be obtained with visual sensors. The authors of [1] account for twenty years of work at the intersection between the robotics and computer vision communities. In many works, as in [2], computer vision techniques are used in a landmark-based framework. Identifying 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 3D model or a map built online. They generally rely on a complete or partial 3D 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 in three steps 1) visual memory building, 2) localization, 3) autonomous navigation.

In the first off-line step (visual memory building), a sequence of images is acquired during a human-guided navigation. It allows us to derive paths driving the vehicle from its initial to its goal locations. In order 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 describes more precisely this point.

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 which best fits the current image in the visual memory. With this aim, we propose a hierarchical process combining 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.
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 in 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 outdoor environment along a 750-meter-long trajectory. Results are presented in Section V.

II. VISUAL MEMORY AND ROUTES

In [1], approaches using 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 $\Psi^p$ is then stored and indexed as the $p^{th}$ learnt path.

1) Visual paths: A visual path $\Psi^p$ is a weighted directed graph composed of $n$ successive key images (vertices):

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

For control purpose (refer to Section IV), the authorized motions during the learning stage 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 $R_i F_i$ and $R_{i+1} F_{i+1}$, respectively associated to the vehicle when two successive key images $I_i$ and $I_{i+1}$ of a visual path $\Psi$ were acquired, there exists an admissible path $\psi$ from $R_i F_i$ to $R_{i+1} F_{i+1}$ for a car-like vehicle whose turn radius is bounded, and which only moves forward.

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

Hypothesis 2.2: Two successive key images $I_i$ and $I_{i+1}$ contain a set $P_i$ of matched visual features, which can be observed along a path performed between $R_i F_i$ and $R_{i+1} F_{i+1}$ 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, the Hypothesis 2.2 constrains the choice of the key images. The key images selection process is detailed in Section II-C. As a consequence of Hypothesis 2.1 and 2.2, each visual path $\Psi^p$ corresponds to an oriented edge which 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: In order to connect two visual 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 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 2.1: Given two visual paths $\Psi^{p_1} = \{T_i^{p_1} | i \in \{1, 2, \ldots, n_1\}\}$ and $\Psi^{p_2} = \{T_i^{p_2} | i \in \{1, 2, \ldots, n_2\}\}$, if the two key images $T_i^{p_1}$ and $T_i^{p_2}$ abide by both Hypothesis 2.1 and 2.2, then a vertex connects $\Psi^{p_1}$ to $\Psi^{p_2}$.

We also assume this Proposition 2.1 in the particular case where the terminal extremity of a visual path $\Psi^{p_1}$ is the same key image as the initial extremity of another visual path $\Psi^{p_2}$. This is useful in practice, when building the visual memory.

3) A connected multigraph of weighted directed graphs: According to Sections II-A1 and II-A2, 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 yet 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.
the neighborhoods of the search region in image center has coordinates point in each image with Harris corner detector [6]. For an interest vehicles in outdoor environment. Interest points are detected successfully applied for the metric localization of autonomous We use a similar process to the one proposed in [3] and necessary input for state estimation (refer to Section IV).

steps of the proposed navigation framework. It allows key useful during autonomous navigation in order to provide the memory which best fits the current image by comparing pre-processed and on-line acquired images. We particularly focus on a method suitable when the data set consists of omnidirectional images. Omnidirectional cameras are usually intended as a vision system providing a huge field-of-view. Such an enhanced field of view can be achieved by either using catadioptric systems, obtained by opportunely combining mirrors and conventional cameras, or employing purely dioptric fish-eye lenses [7]. As first demonstrated in [8] and exploited in robotic applications in [9], images acquired by those sensors have a similar behaviour. In our experiments, a fisheye camera is employed.

The efficiency of a visual localization method can be measured by means of: 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 images: the image can be represented by a single descriptor (global approaches) [10], [11], [12] or alternatively by a set of descriptors defined around visual features (landmarks-based or local approaches) [13], [14], [15]. In those last methods, some relevant visual features are extracted from the images. A descriptor is then associated to each feature neighbourhood. The robustness of the extraction and the invariance of the descriptor are one main issue 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, global descriptors speed up the matching process at the price of affecting the robustness to occlusions. Some approaches are hierarchical [16], [15]: a first selection is done using a global descriptor while 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 which allows to select a set of candidate images. Local descriptors are then exploited to select only the best image and thus to ensure accuracy.

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 which best fits the current image by comparing pre-processed and on-line acquired images. We particularly focus on a method suitable when the data set consists of omnidirectional images. Omnidirectional cameras are usually intended as a vision system providing a huge field-of-view. Such an enhanced field of view can be achieved by either using catadioptric systems, obtained by opportunely combining mirrors and conventional cameras, or employing purely dioptric fish-eye lenses [7]. As first demonstrated in [8] and exploited in robotic applications in [9], images acquired by those sensors have a similar behaviour. In our experiments, a fisheye camera is employed.

The efficiency of a visual localization method can be measured by means of: 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 images: the image can be represented by a single descriptor (global approaches) [10], [11], [12] or alternatively by a set of descriptors defined around visual features (landmarks-based or local approaches) [13], [14], [15]. In those last methods, some relevant visual features are extracted from the images. A descriptor is then associated to each feature neighbourhood. The robustness of the extraction and the invariance of the descriptor are one main issue 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, global descriptors speed up the matching process at the price of affecting the robustness to occlusions. Some approaches are hierarchical [16], [15]: a first selection is done using a global descriptor while 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 which allows to select a set of candidate images. Local descriptors are then exploited to select only the best image and thus to ensure accuracy.

C. Key-images selection

A central clue for implementation of our framework relies on efficient point matching. This process takes places 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 in order to provide the necessary input for state estimation (refer to Section IV). We use a similar process to the one proposed in [3] and successfully applied for the metric localization of autonomous vehicles in outdoor environment. Interest points are detected in each image with Harris corner detector [6]. For an interest point to at coordinates (x, y) in image we define a search region in image The search region is a rectangle whose center has coordinates (x, y). For each interest point to inside the search region in image we compute a score between the neighborhoods of and using a Zero Normalized Cross Correlation. 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 which 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 . A key frame is then chosen so that there are as many video frames as possible between and while there are at least common interest points tracked between and .

### Visual route

A visual route describes the vehicle’s mission in the sensor space. Given two key images of the visual memory and, corresponding respectively to the starting and goal locations of the vehicle in the memory, a visual route is a set of key images which describes a path from to as presented in Figure 2. The closest key image to the current image 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 connecting two vertices associated to and. 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 using Dijkstra’s algorithm. Consequently, the visual route results from the concatenation of indexed visual paths. Given two visual paths and, respectively containing indexed key images, the concatenation operation of and is defined as follows:

\[
\begin{align*}
\Psi^{p_1} \oplus \Psi^{p_2} &= \{ I_j^{p_1}, j = \{1, \ldots, n_1, n_1 + 1, \ldots, n_1 + n_2\} \} \\
I_j^{p_1+p_2} &= \begin{cases} 
I_j^{p_1} & \text{if } j \leq n_1 \\
I_j^{p_2} & \text{if } n_1 < j \leq n_1 + n_2
\end{cases}
\end{align*}
\]

### Cross Correlation

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 which 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 . A key frame is then chosen so that there are as many video frames as possible between and while there are at least common interest points tracked between and .

### 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 which best fits the current image by comparing pre-processed and on-line acquired images. We particularly focus on a method suitable when the data set consists of omnidirectional images. Omnidirectional cameras are usually intended as a vision system providing a huge field-of-view. Such an enhanced field of view can be achieved by either using catadioptric systems, obtained by opportunely combining mirrors and conventional cameras, or employing purely dioptric fish-eye lenses [7]. As first demonstrated in [8] and exploited in robotic applications in [9], images acquired by those sensors have a similar behaviour. In our experiments, a fisheye camera is employed.

The efficiency of a visual localization method can be measured by means of: 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 images: the image can be represented by a single descriptor (global approaches) [10], [11], [12] or alternatively by a set of descriptors defined around visual features (landmarks-based or local approaches) [13], [14], [15]. In those last methods, some relevant visual features are extracted from the images. A descriptor is then associated to each feature neighbourhood. The robustness of the extraction and the invariance of the descriptor are one main issue 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, global descriptors speed up the matching process at the price of affecting the robustness to occlusions. Some approaches are hierarchical [16], [15]: a first selection is done using a global descriptor while 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 which allows to select a set of candidate images. Local descriptors are then exploited to select only the best image and thus to ensure accuracy.
The surface is interpolated by a cubic function. The required displacement relations.

B. First selection and Local descriptor

The global distance between those two images is:

\[ d_{i}^{\text{global}} = \min \{d_{i}^{\text{local}}\} \leq t \]

where the threshold \( t \geq 1 \) allows us to not reject the images which have a distance similar to the minimal distance. The output of this first stage is a small amount of candidate images.

We use then a local approach to select the best candidate since only few images are involved (i.e. in this case the computational cost is low). With this aim, a classical local approach based on the Zero Normalized Cross Correlation (ZNCC) between patches around Harris corners is employed since the computational cost is much lower than SIFT or SURF based approaches whereas similar accuracy is obtained with images corresponding to close viewpoints (refer to [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 \( I_k \) such that \( d_{i}^{\text{local}} = \min (d_{i}^{\text{local}}) \).

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 \( I_k \) to the current initial image \( I_c \). A visual route \( \Psi \) connecting \( I_k \) 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, described in the sequel, the key images of the reference visual route are considered as consecutive checkpoints to reach in the sensor space. The 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 \( I_i \) and \( I_{i+1} \) be two consecutive key images of a given visual route to follow and \( I_c \) be the current image. Let us note \( F_i = (O_i, X_i, Y_i, Z_i) \) and \( F_{i+1} = (O_{i+1}, X_{i+1}, Y_{i+1}, Z_{i+1}) \) the frames attached to the vehicle when \( I_i \) and \( I_{i+1} \) were stored and \( F_c = (O_c, X_c, Y_c, Z_c) \) a frame attached to the vehicle in its current location. Figure 5 illustrates this setup. The origin \( O_c \) of \( F_c \) is on the center rear axle of a car-like vehicle, which moves on a perfect ground plane. The hand-eye parameters \( (i.e. \text{the rigid transformation} \ F_c \text{and the frame attached to the camera}) \) are supposed to be known. According to Hypothesis 2.2, the state of a set of visual features \( P_i \) is known in the images \( I_i \) and \( I_{i+1} \). The state of \( P_i \) is also assumed available in \( I_c \) \( (i.e. P_i \) is in the camera field of view). The task to achieve is to drive the state of \( P_i \) from its current value to its value in \( I_{i+1} \). Let us note \( \Gamma \) a path from \( F_i \) to \( F_{i+1} \). The control strategy consists in guiding \( I_i \) to \( I_{i+1} \) by regulating asymptotically the axle \( Y_c \) on \( \Gamma \). The control objective is achieved if \( Y_c \) is regulated to \( \Gamma \) before the origin of \( F_c \) reaches the origin of \( F_{i+1} \).
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 $\Gamma$, 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 Figure 5).

- $O_C$ is the center of the vehicle rear axle,
- $M$ is the point of $\Gamma$ which is the closest to $O_C$. This point is assumed to be unique which is realistic when the vehicle remains close from $\Gamma$.
- $s$ is the curvilinear coordinate of point $M$ along $\Gamma$ and $c(s)$ denotes the curvature of $\Gamma$ at that point.
- $y$ and $\theta$ are respectively the lateral and angular deviation of the vehicle with respect to reference path $\Gamma$.
- $\delta$ is the virtual front wheel steering angle
- $V$ is the linear velocity along the axle $Y_c$ of $F_c$.
- $l$ is the vehicle wheelbase.

Vehicule configuration can be described without ambiguity by the state vector $(s, y, \theta)$: 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 $\delta$. 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 [19]):

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

Model (1) is clearly singular when $y = \frac{1}{c(s)}$ i.e. when point $O_C$ is superposed with the path $\Gamma$ 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 $\Gamma$.

B. Control Design

The control objective is to ensure the convergence of $y$ and $\theta$ toward 0 before the origin of $F_c$ reaches the origin of $F_{c+1}$. The vehicle model (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, path following control law can be designed and tuned according to celebrated Linear System Theory, while controlling nevertheless the actual nonlinear 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 spatial trajectories can then easily be controlled, whatever the vehicle velocity is [21]
Conversion of the vehicle model (1) into chained form can be achieved thanks to the following state and control transformation:

\[ \Phi([s \ y \ \theta]) = [a_1 \ a_2 \ a_3] \]

\[ \Delta = [s \ y \ (1 - c(s) y) \tan(\theta)] \]

\[ (m_1, m_2) = \Psi(V, \delta) \] (2)

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

\[ \begin{align*}
\dot{a}_1 &= m_1 \\
\dot{a}_2 &= a_3 m_1 \\
\dot{a}_3 &= m_2.
\end{align*} \] (6)

In order 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:

\[ \frac{da_1}{da_1} = \dot{a}_1 \quad \text{and} \quad m_3 = \frac{m_2}{m_1} \]

the chained form (6) can be rewritten:

\[ \begin{align*}
\dot{a}_1' &= 1 \\
\dot{a}_2' &= a_3 \\
\dot{a}_3' &= m_3.
\end{align*} \] (7)

The last two equations of system (7) constitute clearly 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 \((a_2 \ a_3)\). Convergence of these two latter variables can be easily achieved by designing the auxiliary control input as:

\[ m_3 = -K_q a_3 - K_p a_2 \]

\[ (K_p, K_d) \in \mathbb{R}^2 \] (8)

since, reporting (8) in (7), provide with:

\[ \frac{da_2'}{da_1} + K_q a_2' + K_p a_2 = 0. \] (9)

Moreover, since the evolution of the error dynamics (9) is driven by \(a_1 = s\) (distance covered by the vehicle along reference path \( \Gamma \)), 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 \((V \neq 0)\). Control law performances are therefore velocity independent. The study of the second order differential equation (9) can allow us to fix the gains \((K_d, K_p)\) for desired control performances.

The expression of the actual control law \( \delta \) can finally be obtained by inverting the chained transformation:

\[ \delta(y, \theta) = \arctan\left(-l \left[ \cos^3 \theta \left( -K_d(1 - c(s) y) \tan \theta \\
- K_p y + c(s)(1 - c(s) y) \tan^2 \theta + \frac{c(s) \cos \theta}{1 - c(s) y} \right) \right] \right) \] (10)

In our experiments the path to follow is simply defined as the straight line \( \Gamma' = (O_{i+1}, Y_{i+1}) \) (refer to Figure 5). In this case \( c(s) = 0 \) and the control law (10) can be simplified as follows:

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

The implementation of control law (11) requires the on-line estimation of the lateral deviation \( y \) and the angular deviation \( \theta \) of \( F_c \) with respect to \( \Gamma \). 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 \((y, \theta)\) 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 work, we propose to use the unified model described in [23], since it allows to formulate state estimations that are valid for any sensor obeying the unified camera model. In other words, it encompasses all 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 parametrized by \( \xi \) describing the type of sensor and by a matrix \( K \) containing the intrinsic parameters. The coordinates \( x_i \) of the point in the image plane corresponding to the 3D point \( X \) are obtained after three steps:

Step 1 : the world points \( X' \) of coordinates \( X = [X \ Y \ Z]^T \) in the camera frame \( F_m \) are projected onto the unit sphere on a point \( X_m \) of coordinates \( X_m \) in \( F_m \):

\[ X_m = X / \| X \| \]

Step 2 : the point coordinates are then changed to a new reference frame \( F_c \) centered in \( C = (0, 0, -\xi) \) and perspective projected onto the normalized image plane \( Z = 1 - \xi \):

\[ \begin{align*}
X &= \left[ X \ Y \ Z + \xi \| X \| \right] / (Z + \xi \| X \|) \\
Y &= \left[ Z + \xi \| X \| \right] / (Z + \xi \| X \|) \\
C &= 1
\end{align*} \] (12)

Step 3 : finally, the coordinates \( x'_j = [x'_j 1] \) in the image plane are obtained after a plane-to-plane collineation \( K \) of the 2D projective point \( x_i \) \( = K x_i \).

We highlight that \( X_m \) can be computed as a function of the coordinates in the image and the sensor parameter \( \xi \):

\[ X_m = (\eta^{-1} + \xi) x \]

\[ \eta = \left[ x^T \| x \| \right]^T \] (13)
from Ry and the lateral deviation i.e. of the input of the control law (11), scale) can be determined (refer to [27]). Finally, the estimation current image coordinates of the point projected onto the sphere, in the case of the pinhole model, the relation (14) may be written:

\[
\begin{bmatrix}
X_m^T \\
Y_m^T \\
Z_m^T
\end{bmatrix}
= \begin{bmatrix}
X^T \\
Y^T \\
Z^T
\end{bmatrix} = 0
\]  
(14)

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

\[
X_m^T E X_m^{*T} = 0
\]  
(15)

where \( E = R [t]_x \) is the essential matrix [22]. In Equation (15), \( X_m \) (respectively \( X_m^* \)) corresponds to the coordinates of the point projected onto the sphere, in the current image \( I_c \) (respectively in the desired key image). Those coordinates are obtained thanks to the relation (13) and to the coordinates of the point matched in the first and second images. The essential matrix \( E \) between two images can be estimated using five couples of matched points as proposed in [25] if the camera calibration (matrix \( K \)) 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 \( R \) and the translation \( t \) up to a scale) can be determined (refer to [27]). Finally, the estimation of the input of the control law (11), i.e., the angular deviation \( \theta \) and the lateral deviation \( y \), can be computed straightforwardly from \( R \) and \( t \).

**V. EXPERIMENTATIONS**

**A. Experimental set-up**

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

**B. Learning step**

In our experiment, the RobuCab is manually driven along the 800-meter-long path drawn in blue in Fig. 10. This path contains important turns as well as way 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.

![Image 7](image7.png)

Fig. 7. RobuCab vehicle with the embedded camera.

![Image 8](image8.png)

Fig. 8. Some key images of the memory.
C. Localization step and initialisation

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 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 (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 9(a). Given a goal image, a visual route starting from $I^*_i$ and composed of 750 key images has been extracted from the visual memory.

D. Autonomous navigation

The control (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 $V$ is fixed between $1\, m/s^{-1}$ and $0.4\, m/s^{-1}$. $K_p$ and $K_d$ have been set so that the error presents a double pole located at value 0.3. The vehicle successfully follows the learnt path (refer to Fig. 10). The experiment lasts 13 minutes for a path of 754 meters. A mean of 123 robust matches for each frame has been found.

The mean computational time during the online navigation was of 82 ms by image. As can be observed in Fig. 12, the errors in the images decrease to zero until reaching a key image. Lateral and angular errors as well as control input are represented in Fig. 11. As it can be noticed, 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. 11.

Some 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 (refer to Fig. 13(a) and (b) for example) as well as the contents (refer to Fig. 13(i) and (j) where a tram masks many visual features during the autonomous navigation).

Some 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 (refer to Fig. 13(a) and (b) for example) as well as the contents (refer to 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 (Thales Sagitta model). It is accurate to 1 cm (standard deviation) in an horizontal plane when enough satellites are available. The accuracy on a vertical axis is only 20 cm on
DGPS data have been recorded during the learning and the 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. Distances (lateral error) between the vehicle positions during the learning and autonomous stages are reported on 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 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 satisfactory executed (after some images, the vehicle is still at a small distance to the learnt trajectory).

VI. CONCLUSION

We have presented a complete framework for autonomous navigation which enables a vehicle to follow a visual path obtained during a learning stage using a single camera and natural landmarks. The robot environment is modelled 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 which takes...
into account nonholonomic 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 fish eye cameras. Our approach has been validated on an urban vehicle navigating along a 750-meter-long trajectory. The experiments have shown that the navigation strategy to be robust to some changes between the learning and the autonomous navigation steps.

REFERENCES


