Auton Robot (2008) 25: 253–266
DOI 10.1007/s10514-008-9093-8

Indoor navigation of a non-holonomic mobile robot using a visual memory

Jonathan Courbon · Youcef Mezouar · Philippe Martinet

Received: 29 August 2007 / Accepted: 3 June 2008 / Published online: 9 July 2008

Abstract When navigating in an unknown environment for the first time, a natural behavior consists on 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 mobile robots presented in this paper is based on this assumption. During a human-guided learning step, the robot performs paths which are sampled and stored as a set of ordered key images, acquired by an embedded camera. The set of these obtained visual paths is topologically organized and provides a visual memory of the environment. Given an image of one of the visual paths as a target, the robot navigation mission is defined as a concatenation of visual path subsets, called visual route. When running autonomously, the robot is controlled by a visual servoing law adapted to its nonholonomic constraint. Based on the regulation of successive homographies, this control guides the robot along the reference visual route without explicitly planning any trajectory. The proposed framework has been designed for the entire class of central catadioptric cameras (including conventional cameras). It has been validated onto two architectures. In the first one, algorithms have been implemented onto a dedicated hardware and the robot is equipped with a standard perspective camera. In the second one, they have been implemented on a standard PC and an omnidirectional camera is considered.

Keywords Visual navigation · Mobile robot · Central camera · Visual-based control

J. Courbon · Y. Mezouar (✉) · P. Martinet
LASMEA UBP Clermont II, CNRS—UMR6602, 24 Avenue des Landais, 63177 Aubiere, France
e-mail: mezouar@lasmea.univ-bpclermont.fr

1 Introduction

Vision is a central clue of most of recent mobile robots navigation frameworks. The authors of DeSouza and Kak (2002) account of twenty years of works at the meeting point of mobile robotics and computer vision communities. Often used among more "traditional" embedded sensors—proprioceptive sensors like odometers as exteroceptive ones like sonars—it provides accurate localization methods. In many works, and especially those dealing with indoor navigation as in Hayet et al. (2002), computer vision techniques are used in a landmark-based framework. Identifying extracted landmarks to known references allows to update the results of the localization algorithm. These methods are based on some knowledges 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 mobile robot can thus be localized in an absolute reference frame. Both motion planning and robot control can then be designed in this space. The results obtained by the authors of Royer et al. (2004) leave to be forcasted 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 mobile robot with respect to a reference frame.

The principle of this approach is to represent the robot environment with a bounded quantity of images gathered in a set called visual memory. In Remazeilles et al. (2004), this concept is exploited to control the 6 *dof* of a robotic arm under large displacements. A set of images is extracted from a previously learnt database which describes successive targets for a global visual servoing task. In Nierobisch et al. (2007), a similar approach is proposed using SIFT moments.

In these papers, the kinematic constraints of a mobile robot are not considered. In the context of mobile robotics, Matsumoto et al. (1996, 1999) also propose to use a sequence of images, but recorded during a human teleoperated motion, and called View-Sequenced Route Reference. This concept underlines the close link between a human-guided learning stage and the performed paths during an autonomous run. However, the automatic control of the robot is not formulated as a visual servoing task. In this paper, we propose a complete image-based framework (i.e. from environment learning to control) for mobile robots navigation valid for the entire class of central cameras. A sequence of images, acquired during a human-guided learning, allows to derive paths driving the robot 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. The visual memory is structured as a graph which takes into account the environment topology. A navigation task consists then in performing autonomously a visual route which is a concatenation of visual paths. The visual route connects thus in the sensor space the initial and goal configurations. Section 2 details more precisely this point.

The Sect. 3 deals with the vision-based control scheme designed to control the robot motions along a visual route. The nonholonomic constraints of most current wheeled mobile robots make the classical visual servoing methods unexploitable since the camera is fixed on the robot (Tsakiris et al. 1998). However, motivated by the development of 2D 1/2 visual-servoing method proposed by Malis et al. (1999), some authors have investigated the use of homography and epipolar geometry to stabilize mobile robots (Chen et al. 2003; López-Nicolás et al. 2006). In this paper, because the notions of visual route and path are very closed, we turn the nonholonomic visual-servoing issue into a path following one. The designed control law does not need any explicit off-line path planning step. The on-line navigation step of the framework is summarized in Fig. 1.

In Sect. 4, two implementations of this framework are described. In the first one, a perspective camera is embedded on the mobile robot and algorithms have been implemented onto a dedicated hardware. In the second one, they have been implemented on a standard PC and an omnidirectional camera is considered.

2 Visual memory and routes

In DeSouza and Kak (2002), approaches using a "memorization" of images of the environment acquired with an embedded camera are ranked among mapless navigation systems. Indeed, as proposed in Matsumoto et al. (1996) or in Jones et al. (1997), 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 robot. The first step of our framework consists on a learning stage to build the visual memory. The visual memory is structured according to the environment topology to reduce the computational cost.

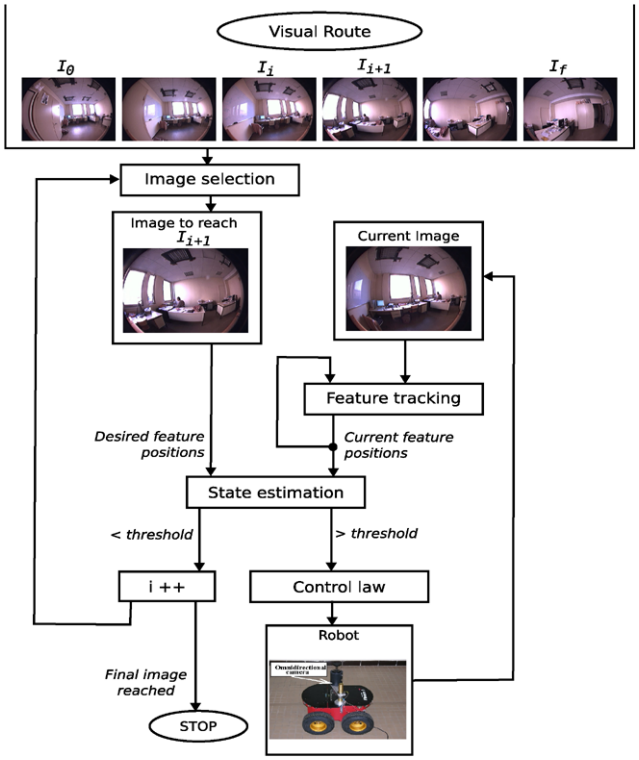

Fig. 1 Principle of the on-line navigation step. The robot is controlled using the current image and the desired image of the visual route extracted from the visual memory (built during a human guided step)

2.1 Structure of the visual memory

The learning stage relies on the human experience. The user guides the mobile robot along one or several paths into each room where the robot is authorized to go (see Fig. 2a, b and c). A visual path ${}^r\Psi_p$ is then stored and indexed as the pth learnt path in the rth room.

2.1.1 Visual paths

A visual path ${}^r\Psi_p$ is a weighted directed graph composed of n successive key images (*vertices*):

$${}^r\Psi_p = \left\{ {}^r\mathcal{I}_i^p | i = \{1, 2, \ldots, n\} \right\}$$

For control purpose (refer to Sect. 3), 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 1 formalizes these constraints.

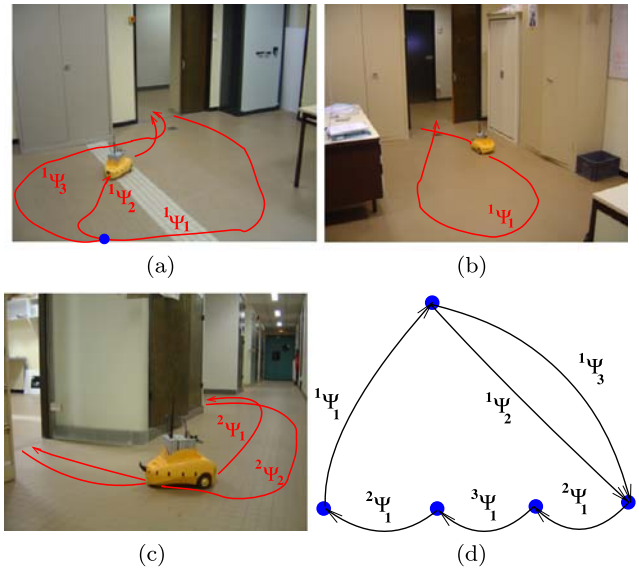

Fig. 2 Building a visual memory: Into the rooms (**a**) and (**b**) and the corridor (**c**), the paths ${}^{r}\Psi_{p}$ have been learnt by teleoperating the robot. As a result, the graph (**d**) represents the topological organization of the visual memory. The blue circles show the vertices

Hypothesis 1 *Given two frames* ${}^{R}\mathcal{F}_i$ *and* ${}^{R}\mathcal{F}_{i+1}$, *respectively associated to the mobile robot 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 which only moves forward.*

Moreover, because the controller is vision-based, the robot is controllable from ${}^{r}\mathcal{I}_i^p$ to ${}^{r}\mathcal{I}_{i+1}^p$ 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 tracked along a path performed between* ${}^{R}\mathcal{F}_i$ *and* ${}^{R}\mathcal{F}_{i+1}$ *and which allows the computation of the control law.*

During the acquisition of a visual path, the Hypothesis 2 constrains the choice of the key images. A new key image $\mathcal{I}_{i+1}$ is stored when a pattern, which has been tracked since $\mathcal{I}_i$ was acquired, is likely to leave the image. A specific pattern tracker based-on particle filtering has been developed for this application. This point is detailled in Sect. 4. As a consequence of Hypothesis 1 and 2, each visual path ${}^{r}\Psi_{p}$ corresponds to an oriented edge which connects two configurations of the robot's workspace. Moreover, the number of key images of a visual path is directly linked to the human-guided path complexity. According to this parameter, we define the *weight of a visual path* as its cardinal.

2.1.2 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 (see Fig. 2d).

Proposition 1 *Given two visual paths* $\Psi_{p_1} = \{\mathcal{I}_i^{p_1} | i = \{1, 2, \ldots, n_1\}\}$ *and* $\Psi_{p_2} = \{\mathcal{I}_i^{p_2} | i = \{1, 2, \ldots, n_2\}\}$, *if the two key images* $\mathcal{I}_{n_1}^{p_1}$ *and* $\mathcal{I}_1^{p_2}$ *abide by both Hypothesis 1 and 2, then a vertex connects* Ψ_{p_1} *to* Ψ_{p_2}.

We also assume this 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.

2.1.3 A connected multigraph of weighted directed graphs

According to Sects. 2.1.1 and 2.1.2, 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 yet necessary that this multigraph is strongly connected. Indeed, this condition warrants that any vertex of the visual memory is attainable from every others, through a set of visual path.

2.2 Visual route

A visual route describes the robot's mission in the sensor space. Given two key images of the visual memory $\mathcal{I}_c^*$ and $\mathcal{I}_g$, corresponding respectively to the current and goal locations of the robot in the memory, a visual route is a set of key images which describes a path from $\mathcal{I}_c^*$ to $\mathcal{I}_g$, as presented in Fig. 3. $\mathcal{I}_c^*$ is the closest key image to the current image acquired by the embedded camera $\mathcal{I}_c$. This can be done in an off-line stage, as Remazeilles et al. propose in Remazeilles et al. (2004), by comparing the photometric invariants of the request image with those of the images stored onto the visual memory. The visual route is the minimum length path of the visual memory connecting two vertices associated to $\mathcal{I}_c$ 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 may be obtained using Dijkstra's algorithm for example. 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 follows:

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

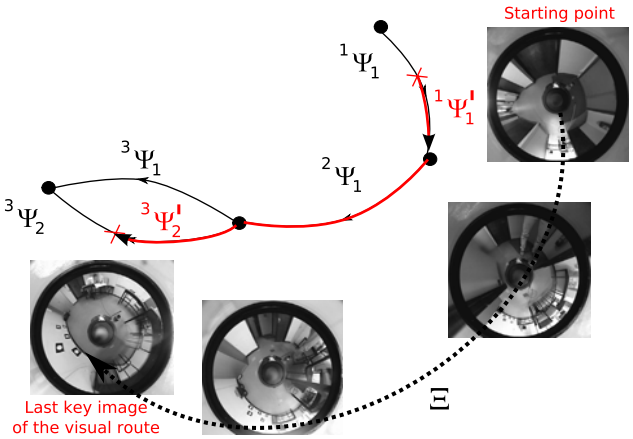

Fig. 3 A visual route: $\Xi = {}^1\Psi'_3 \oplus^2 \Psi_1 \oplus^3 \Psi_1 \oplus^2 \Psi'_1$. ${}^1\Psi'_3$ and ${}^2\Psi'_1$ are subsets of respectively ${}^1\Psi_3$ and ${}^2\Psi_1$. ${}^1\Psi_3$ is splitted at the closest key image to the initial $\mathcal{I}_c$, while the last key image of ${}^2\Psi'_1$ is a desired image to reach by navigating onto the visual memory

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

3 Visual route following

We are considering a sensor-based control strategy. Visual-servoing is often considered as a way to achieve positioning tasks. Classical methods, based on the task function formalism, are based on the existence of a diffeomorphism between the sensor space and the robot's configuration space. Due to the nonholomic constraints of most of wheeled mobile robots, under the condition of rolling without slipping, such a diffeomorphism does not exist if the camera is rigidly fixed to the robot. In Tsakiris et al. (1998), the authors add extra degrees of freedom to the camera. The camera pose can then be regulated in a closed loop.

In the case of an embedded and fixed camera, the control of the camera is generally based on wheeled mobile robots control theory Samson (1995). In Ma et al. (1999), a car-like robot is controlled with respect to the projection of a ground curve in the image plane. The control law is formalized as a path following problem. More recently, in Chen et al. (2003), a partial estimation of the camera displacement between the current and desired views has been exploited to design vision-based control laws. The camera displacement is estimated by uncoupling translation and rotation components of an homography matrix. In Chen et al. (2003), a trajectory following task is achieved. The trajectory to follow is defined by a prerecorded video. In our case, unlike a whole video sequence, we deal with a set of relay images which have been acquired from geometrically spaced out points of view. Indeed, a visual route following can be considered as a sequence of visual-servoing tasks. A stabilization approach could thus be used to control the camera motions from a key image to the next one. However, a visual route is fundamentally a path. 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 set of path following to guide the nonholonomic mobile robot along the visual route.

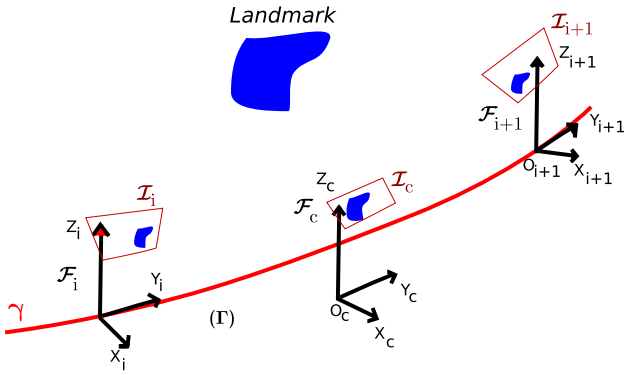

Fig. 4 Frames and images: $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ are two consecutive key images, acquired along a teleoperated path γ and $\mathcal{I}_c$ is the current image

3.1 Assumptions and models

Let $\mathcal{I}_i, \mathcal{I}_{i+1}$ be two consecutive key images of a given visual route to follow and $\mathcal{I}_c$ 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 robot when $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ were stored and $\mathcal{F}_c = (O_c, \mathbf{X}_c, \mathbf{Y}_c, \mathbf{Z}_c)$ a frame attached to the robot in its current location. Figure 4 illustrates this setup. The origin O_c of $\mathcal{F}_c$ is on the axle midpoint of a cart-like robot, which evolutes on a perfect ground plane.

The control vector of the considered cart-like robot is $\mathbf{u} = [V, \omega]^T$ where V is the longitudinal velocity along the axle $\mathbf{Y}_c$ of $\mathcal{F}_c$, and ω is the rotational velocity around $\mathbf{Z}_c$. 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 $\mathcal{I}_{i+1}$. Moreover $\mathcal{P}_i$ has been tracked during the learning step along the path γ between $\mathcal{F}_i$ and $\mathcal{F}_{i+1}$. 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 $\mathcal{I}_c$ to its value in $\mathcal{I}_{i+1}$.

3.2 Control design

Consider the straight line $\Gamma = (O_{i+1}, \mathbf{Y}_{i+1})$ (see Fig. 5). The control strategy consists in guiding $\mathcal{I}_c$ to $\mathcal{I}_{i+1}$ by regulating asymptotically the axle $\mathbf{Y}_c$ on Γ. The control objec-

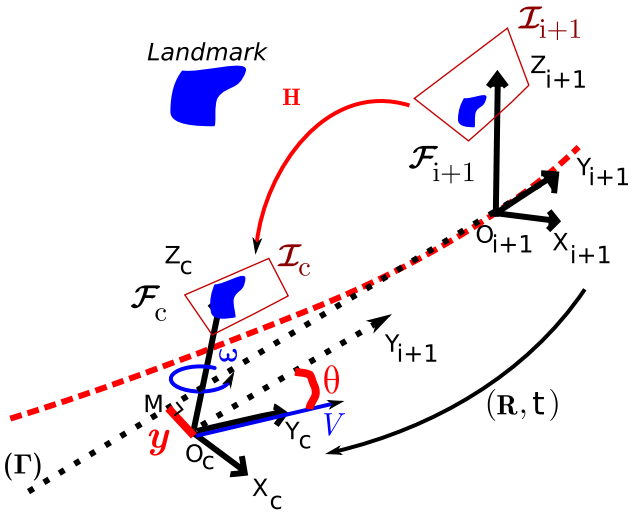

Fig. 5 Control strategy using the homography matrix: the control consists on regulating the lateral error y and the angular error θ to zero

tive is achieved if $\mathbf{Y}_c$ is regulated to Γ before the origin of $\mathcal{F}_c$ reaches the origin of $\mathcal{F}_{i+1}$.

Let us first define $\mathbf{M}$ as the orthogonal projection of the origin of $\mathcal{F}_c$ on Γ and the robot state vectors as $\mathbf{X} = [s\ y\ \theta]^\top$, where s is the curvilinear coordinate of $\mathbf{M}$ along Γ, y is the lateral deviation between the origin of $\mathcal{F}_c$ and Γ, and θ is the angular deviation between $\mathcal{F}_c$ and $\mathcal{F}_{i+1}$. With these notations classical state space model (1) can be calculated:

$$\begin{cases} \dot{s} = V\cos\theta \\ \dot{y} = V\sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{1}$$

The control objectif is to ensure the convergence of y and θ toward 0 before the origin of $\mathcal{F}_c$ reaches the origin of $\mathcal{F}_{i+1}$. This can be done using chained systems since model (1) enters into the class of non-linear systems which can be converted into chained forms (refer for instance to Samson 1995). A chained system results from a conversion of non-linear model into an almost linear. More precisely, injecting into model (1) the non-linear state transformation:

$$\Phi\left(\begin{bmatrix} s & y & \theta \end{bmatrix}\right) = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \triangleq \begin{bmatrix} s & y & \tan(\theta) \end{bmatrix} \tag{2}$$

and describing the motion of the mobile robot with respect to s (instead of with respect to time) leads to the linear model:

$$\begin{cases} \dfrac{d\,a_2}{d\,s} = a_3 \\ \dfrac{d\,a_3}{d\,s} = m_3 \end{cases} \tag{3}$$

Computations show that the new control variable m_3 and the actual control variable ω are related by an invertible transformation. Thanks to linear control theory, it is then possible to design control law m_3 to ensure the convergence of $(a_2\ a_3)$ to 0. In view of (2), this consequently implies the desired convergence of $(y\ \theta)$ to 0. If m_3 is chosen as a classical PD controller, the inversion of the non-linear relation between m_3 and ω gives the non-linear control law to be implemented:

$$\omega(y,\theta) = -V\cos^3\theta K_p y - |V\cos^3\theta| K_d \tan\theta \tag{4}$$

Since control law (4) is designed from system (3), which is driven with respect to curvilinear abscissa, its capabilities are independent of the linear velocity. To be precise, closed loop performance can be adjusted by tuning parameters K_p and K_d which here define a distance of convergence, i.e. the impulse response of y with respect to the covered distance by the point $\mathbf{M}$ on Γ.

The implementation of control law (4) 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 geometrical relationships between two views acquired with a central camera (catadioptric and conventional cameras) are exploited to enable a partial Euclidean reconstruction from which $(y,\ \theta)$ are derived.

3.3 State estimation from two views

One effective way for increasing the field of view of cameras systems is to combine mirrors with conventional imaging system Benosman and Kang (2000). The obtained sensors are referred as catadioptric imaging systems. The resulting imaging systems have been termed central catadioptric when a single projection center describes the world-image mapping. The central catadioptric projection can be modeled by a central projection onto a virtual unitary sphere, followed by a perspective projection onto an image plane. This virtual unitary sphere is centered in the principal effective view point and the image plane is attached to the perspective camera. In this model, called unified model and proposed by Geyer and Daniilidis (2000), conventional perspective camera appears as a particular case. The state estimation process proposed in the sequel is thus valid for central catadioptric camera as well as for conventional perspective cameras.

3.3.1 Camera model

Let $\mathcal{F}_c$ and $\mathcal{F}_m$ be the frames attached to the conventional camera and to the mirror 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 as depicted in Fig. 6). The origins C and M of $\mathcal{F}_c$ and $\mathcal{F}_m$ will be termed optical center and principal projection center respectively. The optical center C has coordinates $[0\ 0\ -\xi]^T$ with respect to $\mathcal{F}_m$ and the image plane $Z = f(\psi - 2\xi)$ is

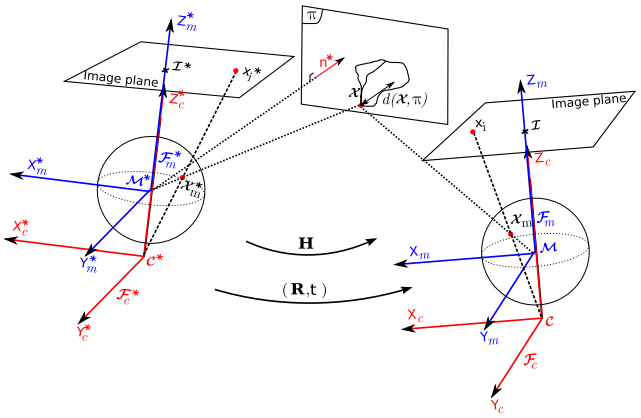

Fig. 6 Unified model for central catadioptric cameras and geometry of two views

orthogonal to the Z-axis where f is the focal length of the conventional camera and ξ and ψ describe the type of sensor and the shape of the mirror, and are function of mirror shape parameters (refer to Barreto and Araujo 2002).

Consider the virtual unitary sphere centered in M as shown in Fig. 6 and let $\mathcal{X}$ be a 3D point with coordinates $\mathbf{X} = [X\ Y\ Z]^T$ with respect to $\mathcal{F}_m$. The world point $\mathcal{X}$ is projected in the image plane into the point of homogeneous coordinates $\mathbf{x_i} = [x_i\ y_i\ 1]^T$. The image formation process can be split in three steps as:

– First step: The 3D world point $\mathcal{X}$ is first projected on the unit sphere surface into a point of coordinates in $\mathcal{F}_m$:

$$\mathbf{X}_m = \frac{\mathbf{X}}{\|\mathbf{X}\|}$$

The projective ray $\mathbf{X}_m$ passes through the principal projection center M and the world point $\mathcal{X}$.

– Second step: The point $\mathbf{X}_m$ lying on the unitary sphere is then perspectively projected on the normalized image plane $Z = 1 - \xi$. This projection is a point of homogeneous coordinates $\underline{\mathbf{x}} = [\mathbf{x}^T\ 1]^T = \mathbf{f}(\mathbf{X})$ (where $\mathbf{x} = [x\ y]^T$):

$$\underline{\mathbf{x}} = \mathbf{f}(\mathbf{X}) = \begin{bmatrix} \dfrac{X}{Z + \xi\|\mathbf{X}\|} & \dfrac{Y}{Z + \xi\|\mathbf{X}\|} & 1 \end{bmatrix}^\top \tag{5}$$

– Third step: Finally the point of homogeneous coordinates $\underline{\mathbf{x}_\mathbf{i}}$ in the image plane is obtained after a plane-to-plane collineation $\mathbf{K}$ of the 2D projective point $\mathbf{x}$:

$$\underline{\mathbf{x}_\mathbf{i}} = \mathbf{K}\underline{\mathbf{x}}$$

The matrix $\mathbf{K}$ can be written as $\mathbf{K} = \mathbf{K}_c\mathbf{M}$ where the upper triangular matrix $\mathbf{K}_c$ contains the conventional camera intrinsic parameters, and the diagonal matrix $\mathbf{M}$ contains the mirror intrinsic parameters:

$$\mathbf{M} = \begin{bmatrix} \psi - \xi & 0 & 0 \\ 0 & \xi - \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{K}_c = \begin{bmatrix} f_u & \alpha_{uv} & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that, setting $\xi = 0$, the general projection model becomes the well known perspective projection model.

In the sequel, we assume that $Z \neq 0$. Let us denote

$$\eta = s\|\mathbf{X}\|/|Z| = s\sqrt{1 + X^2/Z^2 + Y^2/Z^2}$$

where s is the sign of Z. The coordinates of the image point can be rewritten as:

$$x = \frac{X/Z}{1 + \xi\eta}, \qquad y = \frac{Y/Z}{1 + \xi\eta}$$

By combining the two previous equations, it is easy to show that η is the solution of the following second order equation:

$$\eta^2 - (x + y)^2(1 + \xi\eta)^2 - 1 = 0$$

Noticing that the sign of η is equal to the sign of Z, it can be shown that the exact solution is:

$$\eta = \frac{-\gamma - \xi(x^2 + y^2)}{\xi^2(x^2 + y^2) - 1} \tag{6}$$

where $\gamma = \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}$. Equation (6) shows that η can be computed as a function of image coordinates $\mathbf{x}$ and sensor parameter ξ. Noticing that:

$$\mathbf{X}_m = (\eta^{-1} + \xi)\overline{\mathbf{x}} \tag{7}$$

where $\overline{\mathbf{x}} = [\mathbf{x}^T\ \frac{1}{1+\xi\eta}]^T$, we deduce that $\mathbf{X}_m$ can also be computed as a function of image coordinates $\mathbf{x}$ and sensor parameter ξ.

3.3.2 *Scaled Euclidean reconstruction*

Several methods were proposed to obtain Euclidean reconstruction from two views (Faugeras and Lustman 1988). They are generally based on the estimation of the fundamental matrix (Luong and Faugeras 1996) in pixel space or on the estimation of the essential matrix (Longuet-Higgins 1981) in normalized space. However, for control purposes, the methods based on the essential matrix are not well suited since degenerate configurations can occur. Homography matrix and Essential matrix based approaches do not share the same degenerate configurations, for example pure rotational motion is not a degenerate configuration when using homography-based method. The epipolar geometry of central catadioptric system has been more recently investigated (Geyer and Daniilidis 2003; Svoboda et al. 1998).

The central catadioptric fundamental and essential matrices share similar degenerate configurations that those observed with conventional perspective cameras, it is why we will focus on homographic relationship. In the sequel, the collineation matrix $\mathbf{K}$ and the mirror parameter ξ are supposed known. To estimate these parameters the algorithm proposed in Barreto and Araujo (2002) can be used. Let us now show how we can compute homographic relationships between two central catadioptric views.

Let $\mathbf{R}$ and $\mathbf{t}$ be the rotation matrix and the translation vector between two positions $\mathcal{F}_m$ and $\mathcal{F}_m^*$ of the central catadioptric camera (see Fig. 6). Consider a 3D reference plane (π) given in $\mathcal{F}_m^*$ by the vector $\pi^{*\top} = [\mathbf{n}^* \ -d^*]$, where $\mathbf{n}^*$ is its unitary normal in $\mathcal{F}_m^*$ and d^* is the distance from (π) to the origin of $\mathcal{F}_m^*$.

Let $\mathcal{X}$ be a 3D point with coordinates $\mathbf{X} = [X \ Y \ Z]^\top$ with respect to $\mathcal{F}_m$ and with coordinates $\mathbf{X} = [X^* \ Y^* \ Z^*]^\top$ with respect to $\mathcal{F}_m^*$. Its projection in the unit sphere for the two camera positions are:

$$\mathbf{X}_m = (\eta^{-1} + \xi)\overline{\mathbf{x}} = \frac{1}{\rho}\begin{bmatrix} X & Y & Z \end{bmatrix}^\top$$

$$\mathbf{X}_m^* = (\eta^{*-1} + \xi)\overline{\mathbf{x}}^* = \frac{1}{\rho}\begin{bmatrix} X^* & Y^* & Z^* \end{bmatrix}^\top$$

Using the homogenous coordinates $\underline{\mathbf{X}} = [X \ Y \ Z \ H]^\top$ and $\underline{\mathbf{X}}^* = [X^* \ Y^* \ Z^* \ H^*]^\top$, we can write:

$$\rho(\eta^{-1} + \xi)\overline{\mathbf{x}} = \begin{bmatrix} \mathbf{I_3} & 0 \end{bmatrix}\underline{\mathbf{X}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}\underline{\mathbf{X}}^* \tag{8}$$

The distance $d(\mathcal{X}, \pi)$ from the world point $\mathcal{X}$ to the plane (π) is given by the scalar product $\pi^{*\top} \cdot \underline{\mathbf{X}}^*$ and:

$$d(\mathbf{X}^*, \pi^*) = \rho^*(\eta^{*-1} + \xi)\mathbf{n}^{*\top}\overline{\mathbf{x}}^* - d^*H^*$$

As a consequence, the unknown homogenous component H^* is given by:

$$H^* = \frac{\rho^*(\eta^{*-1} + \xi)}{d^*}\mathbf{n}^{*\top}\overline{\mathbf{x}}^* - \frac{d(\mathbf{X}^*, \pi^*)}{d^*} \tag{9}$$

The homogeneous coordinates of $\mathcal{X}$ with respect to $\mathcal{F}_m^*$ can be rewritten as:

$$\underline{\mathbf{X}}^* = \rho^*(\psi^{*-1} + \xi)\begin{bmatrix} \mathbf{I_3} & 0 \end{bmatrix}^\top \overline{\mathbf{x}}^* + \begin{bmatrix} \mathbf{0_{1\times 3}} & H^* \end{bmatrix}^\top \tag{10}$$

By combining (9) and (10), we obtain:

$$\underline{\mathbf{X}}^* = \rho^*(\eta^{*-1} + \xi)\mathbf{A}^*\overline{\mathbf{x}}^* + \mathbf{b}^* \tag{11}$$

where

$$\mathbf{A}_\pi^* = \begin{bmatrix} \mathbf{I_3} & \frac{\mathbf{n}^*}{d^*} \end{bmatrix}^\top \quad \text{and} \quad \mathbf{b}_\pi^* = \begin{bmatrix} \mathbf{0_{1\times 3}} & -\frac{d(\mathcal{X},\pi)}{d^*} \end{bmatrix}$$

According to (11), the expression (8) can be rewritten as:

$$\rho(\eta^{-1} + \xi)\overline{\mathbf{x}} = \rho^*(\eta^{*-1} + \xi)\mathbf{H}\overline{\mathbf{x}}^* + \alpha\mathbf{t} \tag{12}$$

with $\mathbf{H} = \mathbf{R} + \frac{\mathbf{t}}{d^*}\mathbf{n}^{*T}$ and $\alpha = -\frac{d(\mathcal{X},\pi)}{d^*}$.

$\mathbf{H}$ is the Euclidean homography matrix written as a function of the camera displacement and of the plane coordinates with respect to $\mathcal{F}_m^*$. It has the same form as in the conventional perspective case (it is decomposed into a rotation matrix and a rank 1 matrix). If the world point $\mathcal{X}$ belongs to the reference plane (π) (i.e. $\alpha = 0$) then (12) becomes:

$$\overline{\mathbf{x}} \propto \mathbf{H}\overline{\mathbf{x}}^* \tag{13}$$

Note that (13) can be turned into a linear homogeneous equation $\overline{\mathbf{x}} \times \mathbf{H}\overline{\mathbf{x}}^* = \mathbf{0}$ (where $\times$ denotes the cross-product). As usual, the homography matrix related to (π), can thus be estimated up to a scale factor, using four couples of coordinates $(\mathbf{x}_k; \mathbf{x}_k^*)$, $k = 1, \ldots, 4$, corresponding to the projection in the image space of world points $\mathcal{X}_k$ belonging to (π). If only three points belonging to (π) are available then at least five supplementary points are necessary to estimate the homography matrix by using for example the linear algorithm proposed in Malis and Chaumette (2000). From the estimated homography matrix, the camera motion parameters (that is the rotation $\mathbf{R}$ and the scaled translation $\mathbf{t}_{d^*} = \frac{\mathbf{t}}{d^*}$) and the structure of the observed scene (for example the vector $\mathbf{n}^*$) can thus be determined (refer to Faugeras and Lustman 1988).

In our case, the mobile robot is supposed to move on a perfect ground plane. Then the estimation of the angular deviation θ (around the axle Z) between $\mathcal{F}_c$ and $\mathcal{F}_{i+1}$ and of the lateral deviation y (i.e. the distance between the origin of $\mathcal{F}_c$ and Γ) can be extracted straightforwardly from $\mathbf{R} = \mathbf{R}_{i+1}^c$ and $\mathsf{t}/d^* = \mathsf{t}_{i+1}^c/d^*$.

4 Implementations and validations

The proposed framework has been designed for the entire class of central catadioptric cameras (including conventional cameras). It has been validated onto two architectures. In the first one, algorithms have been implemented onto a dedicated hardware and the robot is equipped with a standard perspective camera. In the second one, they have been implemented on a standard PC and an omnidirectional camera is considered.

4.1 Implementations and validations on a dedicated hardware

A central clue for implementation lies on the development of an efficient method to track models of projected patterns onto a video sequence (see Sect. 2). Indeed, the tracker takes

place in all step of the proposed navigation framework. It allows key images selection during the learning stage, of course it is also usefull during the autonomous navigation to provide the necessary input for state estimation (refer to Sect. 3.3). The principle of this tracker as well as its implementation is described in the sequel.

4.1.1 Pattern tracking

Extensive studies of either active or rigid contour tracking have been presented in literature (Isard and Blake 1998; Zhong et al. 2000; Blake et al. 1993; Bascle et al. 1994; Kass et al. 1988). The used methods are usually defined in the Bayesian filtering framework assuming that the evolution of the contour curve state follows a Markov process (evolution model) and that a noisy observation (measurement model) is available. The contour state is tracked using a probabilistic recursive prediction and update strategy (Arulampalam et al. 2002). More recently, Particle filtering was introduced in contour tracking to handle non Gaussianity of noise and non linearity of evolution model (Isard and Blake 1998). The pattern tracker described here is based on contour model and takes into account the image gray scale level variations. It uses the condensation algorithm (Isard and Blake 1998) to track efficiently 2D patterns on cluttered background. An original observation model is used to update the particle filter state.

Problem formulation

The tracking problem can be expressed as the estimation of the state of a dynamical system based on noisy measurements at discrete times. Let us consider that, at time k, the state of a system is defined by a vector $\mathbf{X}_k$ and the measurements by a vector $\mathbf{Z}_k$. Based on a Bayesian approach, the tracking consists in iteratively predicting and updating the posterior density function (pdf) of the system state using respectively the dynamical and the observation models. The pdf $p(\mathbf{X}_k|\mathbf{Z}_{1:k})$ is estimated as the vector $\mathbf{Z}_{1:k} = (\mathbf{Z}_i, i = 1, \ldots, k)$ containing the latest measurements becomes available online.

Particle filtering is an elegant solution in case of nonlinearity of the evolution and measurement functions and non-Gaussianity of noise (Arulampalam et al. 2002). The key idea is to use a Monte Carlo method to represent the pdf of $\mathbf{X}_k$ by a set of samples (particles) $\mathbf{S}_k^{(i)}$. A weight $w_k^{(i)}$ is associated to each particle. It corresponds to the probability of realization of the particle. Starting from a set $\{\mathbf{S}_k^{(i)}, i = 1, \ldots, N_p\}$ of N_p particles with weights $w_k^{(i)}$, the algorithm consists in iteratively: (1) applying the evolution model on each particle to make it evolve toward a new state $\mathbf{S}_{k+1}^{(i)}$, (2) computing the new weights $w_{k+1}^{(i)}$ using the observation model, (3) re-sampling the particle set (particles with small weights are discarded while particles which obtained high scores are duplicated so that N_p remains constant).

Pattern modeling

The pattern model must enable not only real time tracking but also automatic generation and recognition. It must be complex enough to discard ambiguities due to the presence of objects in the background similar to parts of the model and simple enough to reduce the computational. The structure of a pattern is built in two levels: the first level is composed of a set of contours polygonally approximated by segments and arcs, the second level is composed of a list of vectors whose elements represent the evolution of the image gray scales in the gradient direction around point sampled on the contours (refer to Fig. 7). The contour representation is used to automatic generation and recognition and thus to initialize the tracking algorithm. During the tracking, only gray scale vectors are used to estimate the state of the pattern. Let us consider a window of interest in the image with a center (x_c, y_c) and dimensions Δ_x and Δ_y. Each segmented contour is sampled in a set of image points $\{\mathbf{m}^{(j)} = (u^{(j)}, v^{(j)}), j = 1, \ldots, N_m\}$ where N_m is the number of points. At each point, we define a vector $\mathbf{V}^{(j)} = (g_1^{(j)}, g_2^{(j)}, \ldots, g_l^{(j)}, \ldots)^T$ composed of N_s gray scale value samples from the image following the gradient direction at the pixel $\mathbf{m}^{(j)}$ and with a fixed step size δ. $g_l^{(j)}$ is a bilinear approximation of the gray scale values of the nearest four pixels. A pattern model is then defined as follows:

$$\mathbf{M} = \left\{ \left(\mathbf{U}^{(j)}, \mathbf{V}^{(j)}, \mathbf{W}^{(j)} \right), j = 1, \ldots, N_m \right\} \tag{14}$$

with $\mathbf{U}^{(j)} = [x^{(j)}, y^{(j)}, \phi^{(j)}]^T$, where $x^{(j)} = \frac{u^{(j)} - x_c}{\Delta_x}$ and $y^{(j)} = \frac{v^{(j)} - y_c}{\Delta_y}$ are the normalized coordinates of $\mathbf{m}^{(j)}$ inside the interest window and $\phi^{(j)}$ the gradient orientation at $\mathbf{m}^{(j)}$. The vector $\mathbf{W}^{(j)} = (a^{(j)}, b^{(j)}, \ldots)^T$ is composed of a set of parameters defining a function $\tilde{C}_{GG}^{(j)}$ which is an approximation of the one-dimensional discrete normalized auto-correlation function $C_{GG}^{(j)}$ of $G^{(j)}$ where $G^{(j)}(l) = g_l^{(j)}$

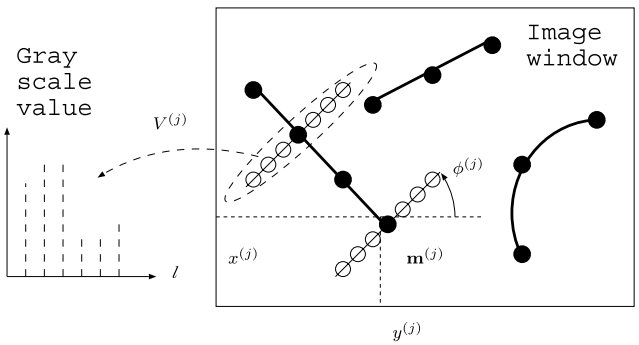

Fig. 7 Pattern model: sampled points on segmented contours and corresponding gray scale vectors in the gradient direction

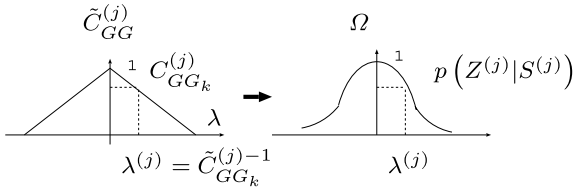

Fig. 8 Deriving the probability measure from the inter-correlation measure

for $l = 1, \ldots, N_s$, and $G^{(j)}(l) = 0$ elsewhere. We have:

$$C_{GG}^{(j)}(\lambda) = \left(1/\parallel \mathbf{V}^{(j)} \parallel^2\right) \sum_{l=-N_S}^{N_S} G^{(j)}(l)\, G^{(j)}(l-\lambda)$$

The simplest expression of $\tilde{C}_{GG}^{(j)}$ is a straight line equation (see Fig. 8). $\mathbf{W}^{(j)}$ is then one-dimensional and equal to the slope.

Observation model

Let us first define the state vector $\mathbf{X}_k$ of a pattern pose in image $\mathcal{I}_k$ at time k as:

$$\mathbf{X}_k = (x_k, y_k, \theta_k, s_k)^T \tag{15}$$

where x_k, y_k, θ_k are respectively the position of the pattern center and its orientation in the image frame and s_k is the scale factor. The evolution of the state vector $\mathbf{X}_k$ is modelled by a noise vector $\mathbf{v}$ from a Gaussian distribution of zero mean value and standard deviation $\sigma_v = (\sigma_x, \sigma_y, \sigma_\theta, \sigma_s)$. Thus, the dynamical model function is $\mathbf{X}_k = \mathbf{X}_{k-1} + \mathbf{v}$.

A key point in particle filtering is the definition of the observation model $\mathbf{Z}_k$ in order to estimate $w_k^{(i)} = p(\mathbf{Z}_k|\mathbf{S}_k^{(i)})$. For each predicted particle $\mathbf{S}_k^{(i)}$, the model is fitted to the image $\mathcal{I}_k$. Around each image point coinciding with a model point $\mathbf{m}_j$, an observed vector $\mathbf{V}_k^{(j)} = (g_1^{(j)}, g_2^{(j)}, \ldots, g_l^{(j)}, \ldots)^T$ of gray scales is built following a direction which is the transformation of the gradient orientation of the model. For each observed vector we compute the normalized inter-correlation between the vector stored in the model and the observed vector components:

$$C_{GG_k}^{(j)} = \frac{\mathbf{V}^{(j)}.\mathbf{V}_k^{(j)}}{\parallel \mathbf{V}^{(j)} \parallel \parallel \mathbf{V}_k^{(j)} \parallel}$$

The question is now how to use the inter-correlation measure to estimate $p(\mathbf{Z}_k|\mathbf{S}_k^{(i)})$? We first compute the probability $p(\mathbf{Z}_k^{(j)}|\mathbf{S}_k^{(i)})$ that each model point $\mathbf{m}^{(j)}$ is placed according to the state vector particle $\mathbf{S}_k^{(i)}$ on the corresponding point in the observed pattern. The maximum of probability is expected at $\mathbf{m}^{(j)}$. The inter-correlation measure $C_{GG^{(k)}}^{(j)}$ can yeld an estimate of the deviation $\lambda^{(j)} = \tilde{C}_{GG}^{(j)-1}(C_{GG_k}^{(j)})$

Fig. 9 The developed prototype with the perspective camera

between the observed and the predicted gray scale vectors. Assuming that the probability that the observed gray scale vector fits the predicted one (Gaussian distribution with respect to $\lambda^{(j)}$ (Fig. 8)), we can reasonably approximate $p(\mathbf{Z}_k^{(j)}|\mathbf{S}_k^{(i)})$ by $\Omega_\sigma(\lambda)$, the one-dimensional Gaussian function with standard deviation σ. Furthermore, assuming that the probabilities $p(\mathbf{Z}_k^{(j)}|\mathbf{S}_k^{(i)})$ are mutually independent, it results that

$$p\left(\mathbf{Z}_k|\mathbf{S}_k^{(i)}\right) = \prod_{j=1}^{N_p} p\left(\mathbf{Z}_k^{(j)}|\mathbf{S}_k^{(i)}\right) \tag{16}$$

The characterization of the auto-correlation function by the parameter $\mathbf{V}_j$ for each model improve the precision of the estimate of $p(\mathbf{Z}_k|\mathbf{S}_k^{(i)})$. Indeed, the intercorrelation may decrease faster for a highly textured contour point than for a point on a contour defined by two large and homogeneous regions.

4.1.2 Implementation

The developed algorithms have been integrated and validated on a small mobile robot (cf. Fig. 9). The hardware architecture should be able to interface easily with the already existing system architecture of the Pekee™ robot from Wany Robotics. It should cope with the computation requirements of the navigation algorithms, while fitting in place in the robot and having a reasonable power consumption. High integration constraints are usually solved by developing dedicated System On Chip chipset such as those that can be found in smartphones, PDA, etc. In this aim, a programmable chip (FPGA) with an embedded hardware processor core has been employed. The programmable logic allows the design of various interfaces, of powerful data management and computation units, while the processor allows flexibility for other tasks needed by the navigation algorithms. Another determining factor is the availability of a prototyping board for the targeted device. Providing flash and dynamic memories, several expansion connectors, this board allow to design, integrate and validate at an early stage the whole navigation algorithms directly on the robot in real

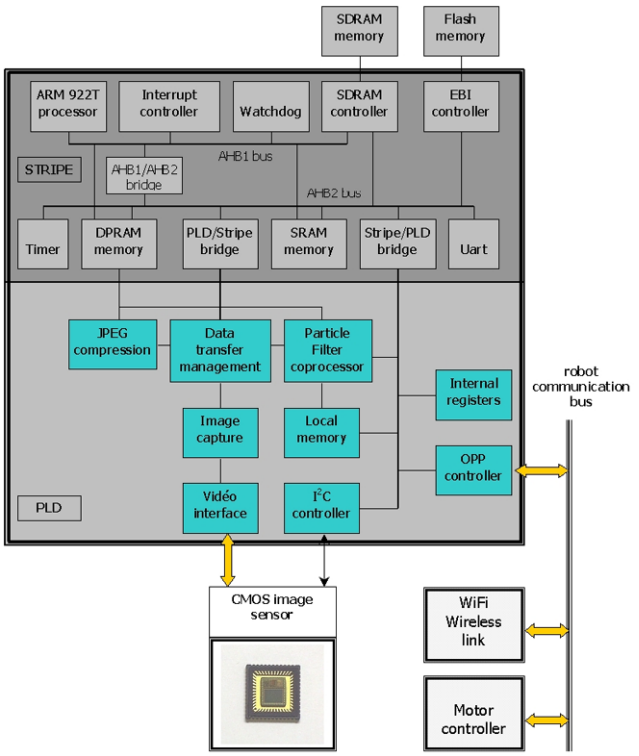

Fig. 10 FPGA main functional hardware blocs

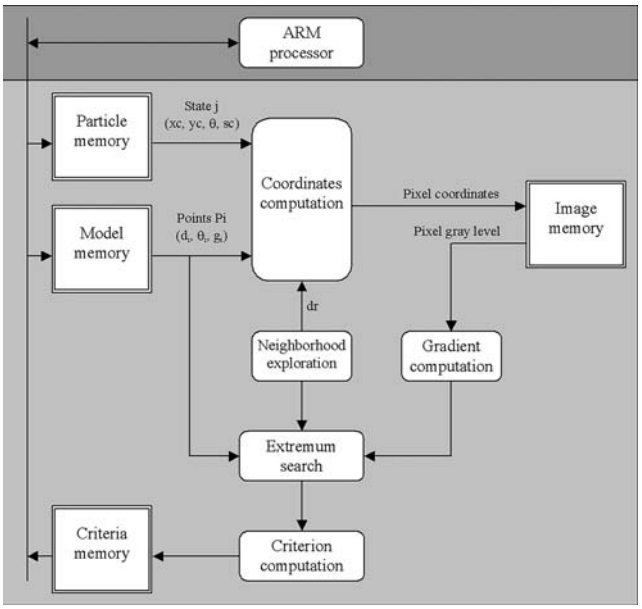

Fig. 11 Block diagram of the particle filter coprocessor

situations. It was then possible to design safely a smaller computational board fitting more efficiently our needs.

An EPXA10 chip from ALTERA was used as the main part of the navigation module. It associates a high density FPGA, an embedded 32 bits RISC ARM9^{TM} processor running at 200 MHz and several peripherals (interrupt controller, dynamic memory controller, timer, UART, ...). The FPGA device drives a CMOS image sensor, performs all the navigation algorithms, and sends speed and steering commands to the robot. The system is completed with a WiFi wireless link already available on the robot essentially used during the learning stage for teleoperation. The main functional blocs of the FPGA are shown in Fig. 10. The perspective camera consists of a single chip $1/3''$ VGA CMOS pixel sensor capturing gray scale images and converting them to a digital raw data stream. While storing this data stream on memory, the FPGA also performs automatic gain and exposure control to adapt in real time for illumination changes. The image sensor is associated with a miniature glass lens offering a 87 diagonal field of view, necessary for covering a wide area. Due to the statistical nature of the particle filter used for the tracking, image distortion does not need to be corrected and the navigation algorithms have been found to work efficiently. Images are used by the particle filter for patterns tracking. Computing requirements for this task are quite heavy and would not fit on the processor alone with the real time constraints of the robot navigation. So most computational part of the particle filter has been implemented in hardware (cf. Fig. 11), relieving the embedded ARM9 processor and insuring good real time performances for the video tracker.

Local memories are used to store the particle state and the planar model to be tracked. A model is defined during the supervised learning step with the polar coordinates of characteristic points corresponding to an extremum of image gradient. For each particle, the corresponding state vector (xc, yc, sc, θ) is applied to the model and the new coordinates of the points are computed. For each predicted point, a radial exploration is performed in the current image to extract the local extremum of the image gradient. The position of this extremum along the radial axis is compared with the considered particle point position. The resulting error is squared and cumulated over each point, providing a measurement criterion for the particle. This step is repeated for all particles and the corresponding criteria are then stored in local memory. All hardware computations are done using fixed point binary representation. Operators are pipelined, and $N \times P \times V$ cycles are needed to fill the criteria memory, where N is the number of particles, P is the number of points defining a planar model and V is number of points for neighborhood exploration. In this application the number of particles is limited to 256 particles, the number of points for a model definition is limited to 64 points, and neighborhood exploration is set to 20 points. With the current logic gate array clocked at 40 MHz, and for the above maximum conditions, this leads to an execution time of 8.2 ms which is compatible with the real time video data processing. The time required for the overall tracking task is 24 ms. It includes image acquisition, particle filter processing and tracker state estimation.

In addition to these specific tasks inherent in tracking algorithm, the FPGA also manages interfaces communication with the others parts of the system. It includes I2C controller and video interface for the image sensor, specific robot communication protocol controller, direct memory data transfer management and jpeg compression. Indeed for monitoring and debug purpose, a wireless link is established between the robot and a remote PC running GUI software. It allows to download configuration for the navigation module and to upload images, tracker state in the image and robot state. As image transfer could required a large amount of data, the jpeg compression, implemented in hardware, is used to reduce the needed bandwidth.

Finally, management of the navigation task is implemented by the ARM9 processor. It consists of providing the particle filter with pattern models that are possibly visible in the current image. Estimation of the model state in the image is then transposed back to estimate the state of the robot. From this estimation, speed and steering trajectory tracking controls are computed (using control law 4) and sent to the motor controller to drive the robot to the next key image. The cycle time for the overall system is about 70 ms, which is compatible with the robot dynamic. The software program size is 92 KB while the hardware implementation uses 26% of the FPGA logic resources, among the 38400 logic blocs available in this device. For the FPGA and the associated memories, power consumption measured during experiments was 2.5 W.

4.1.3 Experimental results

Learning step

Starting from an initial configuration, an image capture is performed by the FPGA and a pattern model is automatically generated. Then the particle filter is initialized with this first pattern. When the robot is moved, new models are generated and tracked to cover the newly explored space. A new key image is stored when the tracked patterns are likely to leave the image. The collection of all these records are used to build the visual memory of the environment and to create visual paths. In this experiment, 80 key images have been stored.

Autonomous step

The reference path, which is represented on the Fig. 12 by the white squares which are lying on the ground, has been acquired as a visual route of fifteen key images. The Fig. 13 illustrates the evolution of planar patterns tracked during the robot motion along the given visual route. At each frame, the tracker provides the coordinates of a current tracked planar pattern. **H** is then computed thanks to the knowledge of this pattern in the key image to reach $\mathcal{I}_{i+1}$. The robot state is then estimated from **H**. A key image is assumed to be reached when the state vector is smaller than a fixed threshold. The longitudinal velocity V was fixed to 0.2 ms^{-1}. When the robot stops at the end of the visual route, the positioning error is around 5 cm and the angular error about 3°. Nevertheless, note that the robot has been stopped roughly, by setting V to zero since the last key image of the visual route has been detected. Moreover, both camera intrinsic and hand-eye parameters has been roughly determined. Errors and control outputs are represented in Fig. 14. The discontinuities are due to the transition between two key images. As it can be noticed, lateral and angular errors are well regulated to zero for each key views.

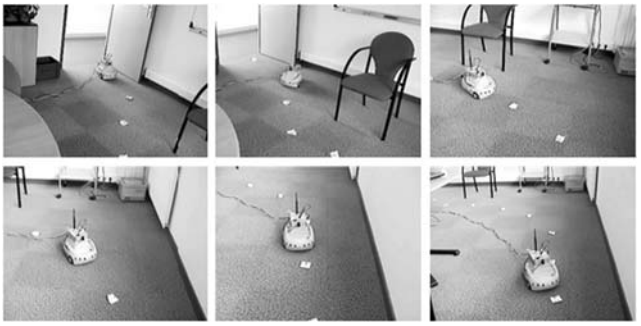

Fig. 12 Following a visual route: the previously learnt visual path, about 10 m long, is materialized on the ground. The pictures were taken during an autonomous run

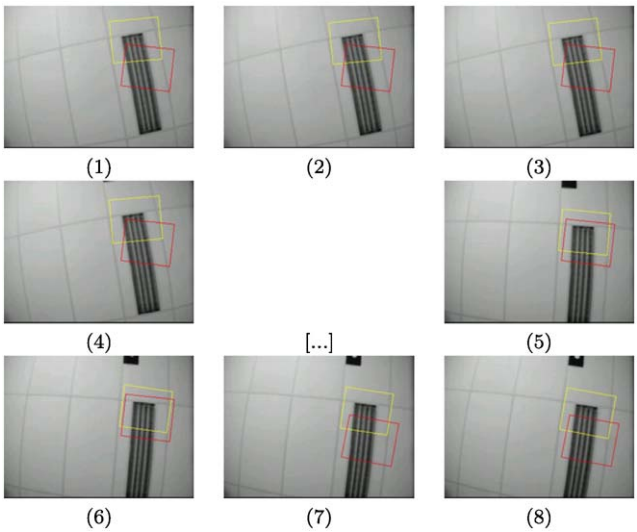

Fig. 13 Evolution of the image space when the robot is regulated between two consecutive key image: in each image, the yellow square is the current state of the tracker, the red one is the state to reach. At image (**7**), a new reference state is given for the tracker. The image (**6**) is thus considered close to the previous reference key image: the control has succeeded

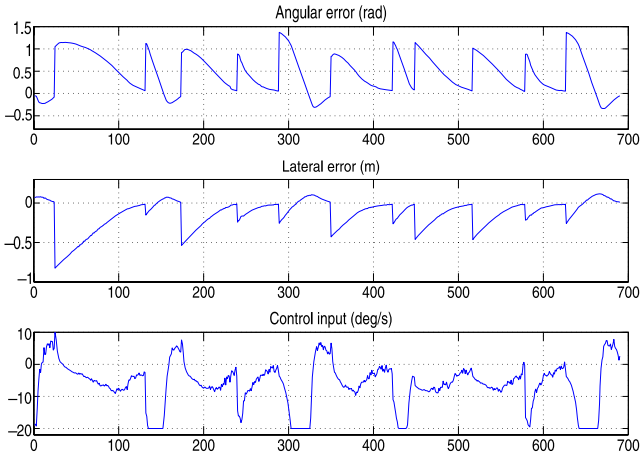

Fig. 14 Evolution of the lateral (in m) and the angular (in rad) errors and of the control input (angular speed in deg/s) during the experimentation with the perspective camera

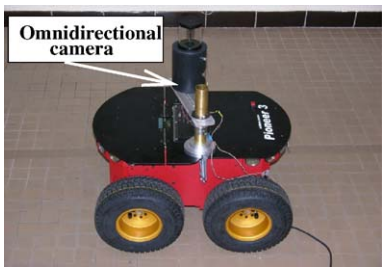

Fig. 15 Mobile robot Pioneer 3 equipped with an omnidirectional camera

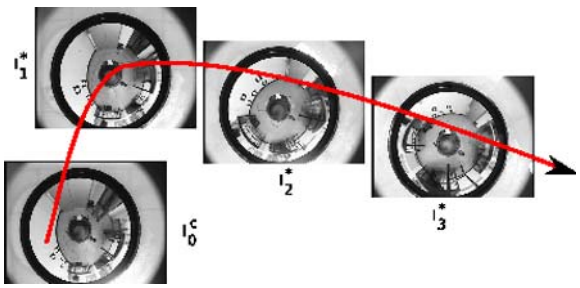

Fig. 16 Visual route with the central catadioptric camera. The current image of the camera is I_c^0. The visual route the robot has to follow is composed of three key images I_1^*, I_2^*, I_3^*

4.2 Experiments with a catadioptric camera

The proposed framework has been also implemented on an external standard PC which wireless controls a Pioneer 3AT robot. An omnidirectional camera is embedded on the robot (see Fig. 15).

A learning stage has been conducted off-line and images have been memorized as proposed in Sect. 2. Three key views (cf. Fig. 16) have been selected to drive the robot from its initial configuration to the desired one (note that only few key images are necessary to cover a large environment with an omnidirectional camera). The results of the experimentation (Fig. 17) confirm once again the validity of our approach: the lateral and the angular errors are regulated to zero before reaching a key image, even with a rough calibration.

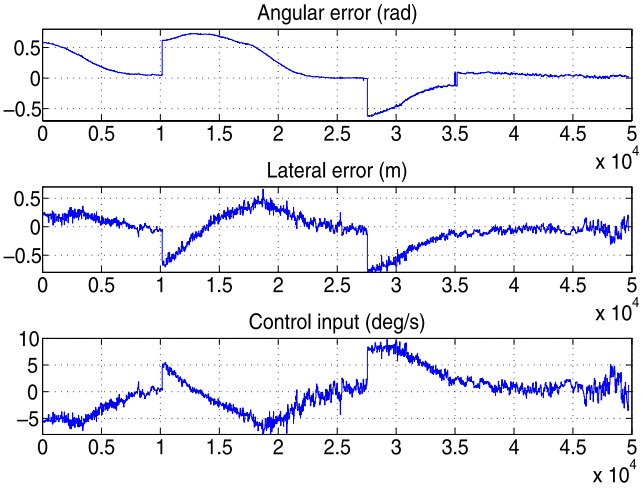

Fig. 17 Evolution of the lateral (in m) and the angular (in rad) errors and of the control input (angular speed in deg/s) during the experimentation with the catadioptric camera

5 Conclusion

This paper described an original framework for image-based navigation dedicated to nonholonomic mobile robots. In this framework, an off-line session allows to learn the environment as a graph of visual paths. The set of these obtained visual paths is topologically organized and provides a visual memory of the environment. Given an image of one of the visual paths as a target, the robot navigation mission is defined as a concatenation of visual path subsets, called visual route and which describes, in the sensor space, an admissible path for the mobile robot. This visual route is then performed thanks to a vision-based control law adapted to the robot nonholonomy. The proposed framework enables autonomous navigation without requiring any absolute geometrical localization of the robot. Furthermore, it has been designed for the entire class of central cameras (including conventional perspective cameras as well as catadioptric sensors). Two implementations of our strategy have been described. The first one combines a dedicated hardware and a standard perspective camera whereas in the second one algorithms have been implemented on a standard PC and an omnidirectional camera has been considered. Future work will be devoted to adequately combine disparate cameras (perspective, catadioptric, fisheye) for navigation tasks.

References

Arulampalam, S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, *50*, 174–188.

Barreto, J., & Araujo, H. (2002). Geometric properties of central catadioptric line images. In *7th European conference on computer vision, ECCV'02* (pp. 237–251). Copenhagen, Denmark.

Bascle, B., Bouthemy, P., Deriche, R., & Meyer, F. (1994). Tracking complex primitives in an image sequence. In *12th international conference on pattern recognition* (pp. 426–431).

Benosman, R., & Kang, S. (2000). *Panoramic vision*. New York: Springer. ISBN 0-387-95111-3.

Blake, A., Curwen, R., & Zisserman, A. A. (1993). A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision*, *11*, 127–145.

Chen, J., Dixon, W. E., Dawson, D. M., & McIntire, M. (2003). Homography-based visual servo tracking control of a wheeled mobile robot. In *International conference on intelligent robots and systems* (pp. 1814–1819). Las Vegas, Nevada.

DeSouza, G. N., & Kak, A. C. (2002). Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(2), 237–267.

Faugeras, O., & Lustman, F. (1988). Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, *2*(3), 485–508.

Geyer, C., & Daniilidis, K. (2000). A unifying theory for central panoramic systems and practical implications. In *European conference on computer vision* (Vol. 29(3), pp. 159–179). Dublin, Ireland.

Geyer, C., & Daniilidis, K. (2003). Mirrors in motion: Epipolar geometry and motion estimation. In *International conference on computer vision, ICCV03* (pp. 766–773). Nice, France.

Hayet, J. B., Lerasle, F., & Devy, M. (2002). A visual landmark framework for indoor mobile robot navigation. In *International conference on robotics and automation (ICRA'02)* (pp. 3942–3947). Washington DC, USA.

Isard, M., & Blake, A. (1998). Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision*, *29*, 5–28.

Jones, S. D., Andersen, C., & Crowley, J. L. (1997). Appearance based processes for visual navigation. In *IEEE/RSJ international conference on intelligent robots and systems* (Vol. 2, pp. 551–557). Grenoble, France.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: active contours. *International Journal of Computer Vision*, *1*, 321–331.

Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, *293*, 133–135.

López-Nicolás, G., Sagüés, C., Guerrero, J. J., Kragic, D., & Jensfelt, P. (2006). Nonholonomic epipolar visual servoing. In *International conference on robotics and automation (ICRA'06)* (pp. 2378–2384).

Luong, Q.-T., & Faugeras, O. (1996). The fundamental matrix: theory, algorithms, and stability analysis. *International Journal of Computer Vision*, *17*(1), 43–76.

Ma, Y., Kosecka, J., & Sastry, S. S. (1999). Vision guided navigation for a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, *15*(3), 521–337.

Malis, E., & Chaumette, F. (2000). 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, *37*(1), 79–97.

Malis, E., Chaumette, F., & Boudet, S. (1999). 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, *15*(2), 238–250.

Matsumoto, Y., Inaba, M., & Inoue, H. (1996). Visual navigation using view-sequenced route representation. In *Proc. of the IEEE international conference on robotics and automation* (Vol. 1, pp. 83–88). Minneapolis, Minnesota.

Matsumoto, Y., Ikeda, K., Inaba, M., & Inoue, H. (1999). Visual navigation using omnidirectional view sequence. In *Int. conf. on intelligent robots and systems* (pp. 317–322).

Nierobisch, T., Krettek, J., Khan, U., & Hoffmann, F. (2007). Optimal large view visual servoing with sets of SIFT features. In *IEEE international conference on robotics and automation, ICRA'07* (pp. 2092–2097).

Remazeilles, A., Chaumette, F., & Gros, P. (2004). Robot motion control from a visual memory. In *IEEE int. conf. on robotics and automation, ICRA'04* (Vol. 4, pp. 4695–4700). New Orleans.

Royer, E., Lhullier, M., Dhome, M., & Chateau, T. (2004). Towards an alternative GPS sensor in dense urban environment from visual memory. In *British machine vision conference* (Vol. 1, pp. 197–206). Kingston, England.

Samson, C. (1995). Control of chained systems. application to path following and time-varying stabilization of mobile robots. *IEEE Transactions on Automatic Control*, *40*(1), 64–77.

Svoboda, T., Pajdla, T., & Hlavac, V. (1998). Motion estimation using central panoramic cameras. In *IEEE conference on intelligent vehicles* (pp. 335–340). Stuttgart, Germany.

Tsakiris, D., Rives, P., & Samson, C. (1998). Extending visual servoing techniques to nonholonomic mobile robots. In G. Hager, D. Kriegman, & A. Morse (Eds.), *LNCIS: Vol. 237. The confluence of vision and control* (pp. 106–117). Berlin: Springer.

Zhong, Y., Jain, A. K., & Dubuisson, M. P. (2000). Object tracking using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 544–549.

Jonathan Courbon is graduated from Institut Français de Mécanique Avancée (IFMA), Clermont-Ferrand (France), in 2005. He is now pursuing his Ph.D. at the Robotics and Vision Group of LASMEA-CNRS, Clermont-Ferrand. His research focuses on the use of visual informations for 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 has been Postdoctoral Associate in the Robotics Laboratory, Computer Science Department, Columbia University, New York, NY, and currently holds a position as Associate Professor at the Blaise Pascal University (France). His research interests include robotics, microrobotics, computer vision, and vision-based control.

Philippe Martinet graduated from the CUST, Clermont-Ferrand (France), in 1985 and received the Ph.D. degree in electronics science from the Blaise Pascal University, Clermont-Ferrand (France), in 1987. From 1990 til 2000, he was assistant Professor with CUST in the Electrical Engineering Department, Clermont-Ferrand. Since 2000, he has been a Professor with Institut Français de Mécanique Avancée (IFMA), Clermont-Ferrand. He was the leader of the group GRAVIR (over 74 people) from 2001 to 2006. He is performing research at the Robotics and Vision Group of LASMEA-CNRS, Clermont-Ferrand. His research interests include visual servoing, force-vision coupling, multi-sensor based control, autonomous vehicles, and modeling and control of complex machines.