Indoor Navigation of Mobile Robot: An Image Based Approach

G Blanc, O Aïder, Y. Mezouar, T. Chateau, Philippe Martinet

Indoor Navigation of Mobile Robot: An Image Based Approach

G.Blanc, O.Aït Aïder, Y.Mezouar, T.Chateau and P. Martinet

LASMEA
24 avenue des Landais
63177 AUBIERE - FRANCE
gblanc,aitaider,mezouar,chateau,martinet@lasmea.univ-bpclermont.fr

Abstract

In this paper, an image-based framework for navigation of a mobile robot in an indoor environment is presented. The only sensor used is an embedded monocular vision system. The environment is autonomously mapped during a learning stage in order to locate the robot on-line. A suitable control law is designed to drive the robot in an image database. To address this issue, a Virtual NonHolonomic Vehicle (VNHV) attached to the image plane is defined.

Keywords

Mobile robot, Image-based navigation, nonholonomic visual servoing.

1 Introduction

In mobile robotics, a complete navigation system must couple environment perceiving and understanding abilities from one hand and control functionality from the other hand. For an indoor mobile robot system, the ability to autonomously map its environment and localize itself relatively to this map is a highly desired property. Using natural rather than artificial landmarks is another important requirement. Several works using monocular vision for mapping and self localization exist. The most important difficulty is to achieve the generation of a sufficient number of landmarks which can be robustly recognized during navigation session with a near real time rate. Interest points ([9]), straight lines ([10], [8], [1]) and rectangular patterns ([3]) where used. The first approaches focused on producing efficient algorithms to match a set of observed patterns with a subset of the map primitives ([10], [8], [1]). Generally, fusion of multiple sensor data (odometry) was used to achieve real time computing and ambiguity eliminating. More recently, the success of real-time tracking algorithms simplified the matching process and allowed to use structure from motion technics to compute de 3D coordinates of the observed features ([9]).

The control module must enable the mobile robot to reach a desired configuration from a current one. Although the final goal of the motion control module is to bring the mobile robot from an initial image to a goal one, the central point of this module lies in controlling the robot motion between two images. This kind of regulation task is often formulated in the robot perception space, using especially the task function approach. In the case of controlling an holonomic robot, like most of manipulator robots, this approach has been successfully used, in particular for visual servoing: time variations of geometrical measures, extracted from the vision system, are linked with the kinematic screw of the robot in its configuration space. Unfortunately, intrinsic constraints on available instantaneous speeds for a nonholonomic robots do not allow such an input kinematic screw to be executed. In the literature, we met two main solutions to this issue. The first one is to increase the dimension of the configuration space of the robot, adding extra degrees of freedom to the sensor ([12]). In [4], the embedded vision system is mounted on pan-tilt platform. The authors propose an image-based control scheme, by defining elementary tasks like wall following and door stabilization. In the second method, the controlled degrees of freedom of the robot are uncoupled to achieve the specified task. In path following application for instance, longitudinal speed is considered just as a parameter of the designed control law for the rotational speed of the vehicle ([11]). Considering only an embedded monocular vision system looking at the ground, Yi Ma et *al* in [7] propose a theoretical framework to regulate the orientation of a wheeled mobile robot to track a ground curve, by expressing a state vector only with data extracted from the projection of the curve in the image.

In this paper, we describe a method for autonomous environment mapping, localization and navigation for an in-

door mobile robot using monocular vision and multiple 2D pattern tracking. The environment map is a set of 2D landmarks (mosaic) on the ceiling plane. The camera is oriented vertically toward the ceiling and is not necessarily calibrated. Visual data are directly used in the control module. we show that the image-based control of a nonholonomic robot can be done by assigning to visual landmarks a nonholonomic behavior. By giving them a longitudinal speed as an input parameter, we regulate the orientation of these landmarks to join and follow a straight line, which is parameterized by their desired attitude. Thanks to a judicious choice of the set of relay images, two successive relay images contain common landmarks, which can be used as input to the control law.

The paper is organized in two distinctive parts. The first part (section 2) deals with the construction of the mosaic of 2D patterns and the pose computing using these patterns. Trajectory learning using key images is also presented. In the second part (section 3), a control law which enable a nonholonomic robot to follow learned paths is described. Experimental validation and simulation results are presented in section 4.

2 Mapping and localization

In this section, we present the method used for mapping and localizing our robot using local 2D landmarks detected on the ceiling of the indoor environment. The map of the ceiling is represented under the form of a mosaic of 2D models. The mosaic is built during a learning stage. Online localization is achieved by recognizing and tracking the 2D models using a particle filter based tracker ([2]). In the following subsections, we will first formalize the positioning problem assuming that the mosaic of the ceiling already exists and then, we will detail the process of map construction.

2.1 On-line robot pose computing

Let $\mathcal{F}_M$ be a 2D frame which lies to the ceiling plane. We assume that a set of 2D landmarks $\mathbf{m_i}$ detected on this plane are modeled and grouped in a mosaic represented by a set $\mathcal{M} = \left\{ \left(\mathbf{m_i}, {}^{\mathbf{M}}\mathbf{T_i} \right), i = 1, ..., n \right\}$ where the planar transformation matrix ${}^{\mathbf{M}}\mathbf{T_i}$ between $\mathcal{F}_M$ and a frame $\mathcal{F}_i$ lied to $\mathbf{m_i}$ defines the pose of $\mathbf{m_i}$ in the mosaic (figure 1). We have:

$$ {}^{\mathbf{M}}\mathbf{T_i} = \begin{bmatrix} {}^{\mathbf{M}}\mathbf{R_i} & {}^{\mathbf{M}}\mathbf{P_i} \\ \mathbf{0} & 1 \end{bmatrix} $$

where ${}^{\mathbf{M}}\mathbf{R_i} = \begin{bmatrix} C\theta_i & -S\theta_i \\ S\theta_i & C\theta_i \end{bmatrix}$ expresses the rotation of the model and ${}^{\mathbf{M}}\mathbf{P_i} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$ its position.

For simplicity, we consider, without loosing generality, that the camera frame is equal to the robot frame. Under this assumption, localizing the robot at the instant k consists in computing ${}^{\mathbf{M}}\mathbf{T}_{\mathbf{C}}^{(\mathbf{k})}$ which defines the homogenous transformation between the projection $\mathcal{F}_C^{(k)}$ of the camera frame $\mathcal{F}_C$ on the mosaic plane as shown in figure 1. This will be achieved using the visual information of the observed landmarks.

At the instant k, the robot grabs an image $\mathbf{I_k}$ of the ceiling. Let $\mathcal{F}_I^{(k)}$ be a 2D frame lied to the image plane. The pose of the projection of an observed model $\mathbf{m_i}$ on the image plane is defined by the transformation ${}^{\mathbf{I}}\mathbf{L}_{\mathbf{i}}^{(\mathbf{k})}$ between $F_I^{(k)}$ and a frame $F_i^{(k)}$ lied to the projection of $\mathbf{m_i}$:

$$ {}^{\mathbf{I}}\mathbf{L}_{\mathbf{i}}^{(\mathbf{k})} = \begin{bmatrix} {}^{\mathbf{I}}\mathbf{r}_{\mathbf{i}}^{(\mathbf{k})} & {}^{\mathbf{I}}\mathbf{p}_{\mathbf{i}}^{(\mathbf{k})} \\ \mathbf{0} & 1 \end{bmatrix} \quad (1) $$

where ${}^{\mathbf{M}}\mathbf{r_i} = \begin{bmatrix} C\theta_i^{(k)} & -S\theta_i^{(k)} \\ S\theta_i^{(k)} & C\theta_i^{(k)} \end{bmatrix}$ express the rotation of the model and ${}^{\mathbf{M}}\mathbf{p_i} = \begin{bmatrix} u_i^{(k)} \\ v_i^{(k)} \end{bmatrix}$ its position.

Considering the inverse of the perspective projection, we obtain the transformation between the projections $F_C^{(k)}$ and F_i of the camera frame and the model frame respectively on the mosaic plane (figure 1):

$$ {}^{\mathbf{C}}\mathbf{T}_{\mathbf{i}}^{(\mathbf{k})} = \begin{bmatrix} {}^{\mathbf{C}}\mathbf{R_i} & {}^{\mathbf{C}}\mathbf{P_i} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^{\mathbf{I}}\mathbf{r}_{\mathbf{i}}^{(\mathbf{k})} & \frac{Z}{f}{}^{\mathbf{I}}\mathbf{p}_{\mathbf{i}}^{(\mathbf{k})} \\ \mathbf{0} & 1 \end{bmatrix} $$

where Z is the distance from the origin of the camera frame to the ceiling and f the focal length.

We can thus express the robot pose relatively to $\mathcal{M}$ with respect to the parameters of a seen model $\mathbf{m_i}$ as follows :

$$ {}^{\mathbf{M}}\mathbf{T}_{\mathbf{C}}^{(\mathbf{k})} = {}^{\mathbf{M}}\mathbf{T_i}\,{}^{\mathbf{i}}\mathbf{T}_{\mathbf{C}}^{(\mathbf{k})} \quad (2) $$

where ${}^{\mathbf{i}}\mathbf{T}_{\mathbf{C}}^{(\mathbf{k})} = \left({}^{\mathbf{C}}\mathbf{T}_{\mathbf{i}}^{(\mathbf{k})} \right)^{-1}$.

The state ${}^{\mathbf{I}}\mathbf{L}_{\mathbf{i}}^{(\mathbf{k})}$ of the model $\mathbf{m_i}$ in the image $\mathbf{I_k}$ is provided thanks to the particle filter tracker ([2]). This tracker provides a criterium w_i which expresses the quality of the state estimate. Suppose that n models $\mathbf{m_i}$ are seen in an image $\mathbf{I_k}$. Each tracker τ_i provides an estimate pose vector $\mathbf{X}_{\mathbf{i}}^{(\mathbf{k})} = \left[x_i^{(k)}, y_i^{(k)}, \theta_i^{(k)} \right]^T$ with a quality factor $w_i^{(k)}$ such that

$$ {}^{\mathbf{M}}\mathbf{T}_{\mathbf{C}}^{(\mathbf{k})} = \begin{bmatrix} C\theta_i^{(k)} & -S\theta_i^{(k)} & x_i^{(k)} \\ S\theta_i^{(k)} & C\theta_i^{(k)} & x_i^{(k)} \\ 0 & 0 & 1 \end{bmatrix} $$

The final robot pose is then equal to the weighted mean value of the poses:

$$ \mathbf{X}^{(\mathbf{k})} = \frac{\Sigma_i w_i^{(k)} \mathbf{X}_{\mathbf{i}}^{(\mathbf{k})}}{\Sigma_i w_i^{(k)}} \quad (3) $$

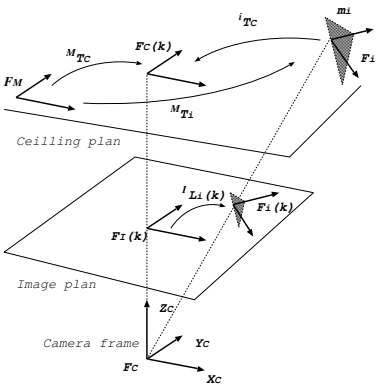


Figure 1: Robot pose computing using visual data of a 2D model

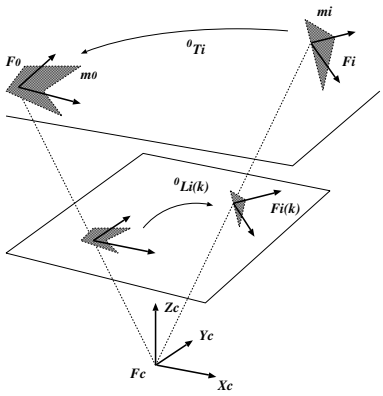


Figure 2: Computing the pose of a new model in the mosaic

Note that the robot position is computed up to a scale factor $\frac{Z}{f}$. The absolute position can be retrieved if the camera is calibrated. Otherwise, the computed pose is sufficient to achieve navigation using visual servoing as we will see in section 2.

2.2 Environment mapping

We will now explain the process of building the mosaic of 2D landmarks. At time $k = 0$, the robot grabs an image I_0, generates a first 2D model $\mathbf{m_0}$ and associates to it a frame $\mathcal{F}_0$. In fact, this first model will serve as a reference to the mosaic. We have thus $\mathcal{F}_0 = \mathcal{F}_M$ and ${}^{\mathbf{M}}\mathbf{T}_{\mathbf{0}}^{(\mathbf{k})} = \mathbf{I}$ ($\mathbf{I}$ is the identity matrix). In the image $\mathbf{I_0}$, a tracker τ_0 is initialized with a state ${}^{\mathbf{I}}\mathbf{L}_{\mathbf{i}}^{(\mathbf{0})}$. As the robot moves, the state of τ_0 evolutes with respect to k. The system generates other models. At each generation of a new model $\mathbf{m_i}$ at the instant k, a new tracker τ_k is initialized with a state ${}^{\mathbf{I}}\mathbf{L}_{\mathbf{i}}^{(\mathbf{k})}$. Due to the mosaic rigidity, the transformation between the two model frames is time independent and equal to ${}^{\mathbf{0}}\mathbf{L_i} = {}^{\mathbf{0}}\mathbf{L}_{\mathbf{I}}^{(\mathbf{k})}{}^{\mathbf{I}}\mathbf{L}_{\mathbf{i}}^{(\mathbf{k})}$ (figure 2). Let us consider that

$$ {}^{\mathbf{0}}\mathbf{L_i} = \begin{bmatrix} {}^{\mathbf{0}}\mathbf{r_i} & {}^{\mathbf{0}}\mathbf{p_i} \\ \mathbf{0} & 1 \end{bmatrix} $$

Projecting this transformation onto the mosaic plane provides the pose of the new model m_i in $\mathcal{M}$:

$$ {}^{\mathbf{M}}\mathbf{T_i} = \begin{bmatrix} {}^{\mathbf{0}}\mathbf{r_i} & \frac{Z}{f}{}^{\mathbf{0}}\mathbf{p_i} \\ \mathbf{0} & 1 \end{bmatrix} \quad (4) $$

Of course, as the robot moves, $\mathbf{m_0}$ may disappear at the instant k when creating a new model $\mathbf{m_i}$. In fact, it is sufficient that at least one model $\mathbf{m_j}$, already defined in the mosaic, is seen at the instant k. To compute the pose ${}^{\mathbf{M}}\mathbf{T_i}$, we first calculate ${}^{\mathbf{j}}\mathbf{L_i}$ and project it to obtain ${}^{\mathbf{j}}\mathbf{T_i}$. The model $\mathbf{m_i}$ is then added to $\mathcal{M}$ with the pose ${}^{\mathbf{M}}\mathbf{T_i} = {}^{\mathbf{0}}\mathbf{T_j}{}^{\mathbf{j}}\mathbf{T_i}$.

2.3 Path learning and visual navigation

The aim of this application is to achieve navigation based exclusively on visual servoing. Since obstacles on the flour are not seen, the idea is to enable the robot to learn and reproduce some paths relating important places in the environment. Let us consider a trajectory executed during the learning phase and relating a point A to a point B. A set of so called key images is chosen among the sequence of video images acquired during navigation following the trajectory. A key image I_k is defined by a set of mosaic models and their poses in the image (figure 3):

$$ I_k^{(AB)} = \left\{ \left(\mathbf{m_i}, {}^{I}L_i^{(k)} \right), i = 1, 2, ... \right\} $$

A trajectory Φ_{AB} relating A and B is then expressed as follows:

$$ \Phi_{AB} = \left\{ I_k^{(AB)}, k = 1, 2, ... \right\} $$

Key images are chosen so that the combination of the elementary trajectories between each couple of successive key images $I_k^{(AB)}$ and $I_{k+1}^{(AB)}$ forms a global trajectory which fits relatively the learned trajectory. Some conditions have to be satisfied when creating Φ_{AB}:

- two successive key images must contain at least one common model of the mosaic,
- the variation of the orientation of a model between two successive key images is smaller than a defined threshold.

The first condition is naturally necessary to visual servoing. The second one is less obvious. It is motivated by the fact that several different paths relating two poses exist when the orientation variation is important. Since the goal is to reproduce a learned trajectory as accurately as possible, it is necessary to insert additional key images to reduce the variation of orientation between two successive images

(figure 3).
The next section focuses on the control law carrying the robot from a key image to another tacking into account nonholonomic constraints.

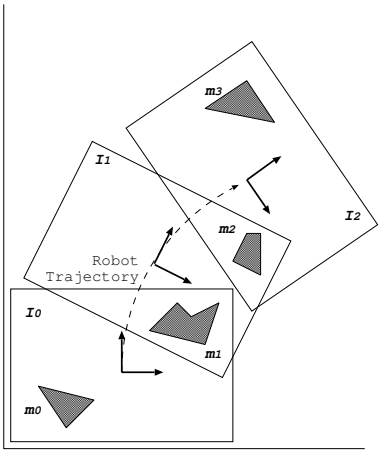


Figure 3: An example of key images forming a trajectory

3 Image-based control law

According to the description done in section 1, we consider that the camera frame $\mathcal{F}_C$ is confounded with the control frame $\mathcal{F}_m$ attached to the mobile robot. This canonical configuration can be extended to cases where a constant rigid transformation, due to extrinsic calibration parameters of the camera, exists between $\mathcal{F}_C$ and $\mathcal{F}_m$. Because we assume that the ceiling is an horizontal plane Π, an homography with respect to Π can be computed between general and canonical images. In the considered configuration, the camera optical axis coincides with the rotation axis of the mobile robot and the camera optical center is confounded with the axle midpoint of the mobile robot. As a consequence, the kinematic constraints of the mobile robot, expressed in $\mathcal{F}_m$, are directly assigned to the camera frame. Then the kinematic screw of the camera is equivalent to a linear velocity along the x-axis of the frame attached to the image plane $\mathcal{F}_O$ and an angular velocity about the optical axis. The robot nonholonomic constraints can thus be assigned to visual landmarks lying in the image plane. We represent them as a "virtual nonholonomic vehicle" (denoted VNHV in the sequel) operating in the image frame $\mathcal{F}_O$ (see Fig. 4).

3.1 Modelisation for control law design

Let $\mathbf{R}$ and $\mathbf{t}$ be the planar rotation and translation between the projections of $\mathcal{L}$ in the current image I_k and desired image I_k*, and define the state vector of a visual landmark in the current and desired images as $\mathbf{X}_\mathcal{L} = \begin{bmatrix} u & v & \varphi \end{bmatrix}^\top$ and $\mathbf{X}^*_\mathcal{L} = \begin{bmatrix} u_* & v_* & \varphi^* \end{bmatrix}^\top$ respectively. These parameters are extracted from the environment map (see section 2.2). Using equation (1), $\mathbf{R}$ and $\mathbf{t}$ are given by:

$$\mathbf{R} = {}^{\mathrm{I}}\mathbf{r}_\mathbf{i}^{(\mathbf{k})}{}^{\mathrm{I}}\mathbf{r}_\mathbf{i}^{(\mathbf{k}*)\top} \quad \mathbf{t} = {}^{\mathrm{I}}\mathbf{p}_\mathbf{i}^{(\mathbf{k})} - \mathbf{R}{}^{\mathrm{I}}\mathbf{p}_\mathbf{i}^{(\mathbf{k}*)}$$

Clearly, driving $\mathbf{X}_\mathcal{L}$ to $\mathbf{X}^*_\mathcal{L}$ is equivalent to drive the VNHV according to the rigid transformation ${}^O\mathbf{T}_{O*} = (\mathbf{R}, \mathbf{t})$. Since ${}^O\mathbf{T}_{O*}$ can be computed from image data, it is possible to plan an adequate path Γ in the image space driving the VNHV to a desired state in the image plane ([5], [6]). The most available and simplest path is a straight line defined in the image space by the position and the orientation to be achieved by the VNHV. These parameters are given directly by ${}^O\mathbf{T}_{O*}$, which is computed at each image acquisition. In the reminder of this paper, we consider only this kind of path. Of course, this modelisation can be extended to other kinds of path, expecially paths which are $\mathcal{C}^2$. Let $\mathbf{M}$ be the closest point on Γ to the center of the cart rear axle $\mathbf{O}$, θ_v the orientation of the cart in the image plane and θ_Γ the orientation of the main vector of Γ. Therefore, $\theta = \theta_\Gamma - \theta_v$ is the angular deviation of the cart with respect to path Γ. The state of the VNHV can be described by a 3 dimensional vector:$\mathbf{X}_v = \begin{bmatrix} s & y & \theta \end{bmatrix}^\top$, where s is the curvilinear coordinate of point $\mathbf{M}$ and y and θ is the lateral deviation with respect to Γ (refer to Fig. 5).

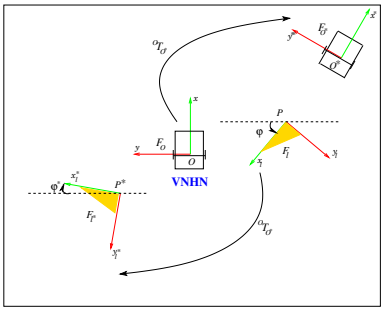


Figure 4: Current and desired configurations in the image plane

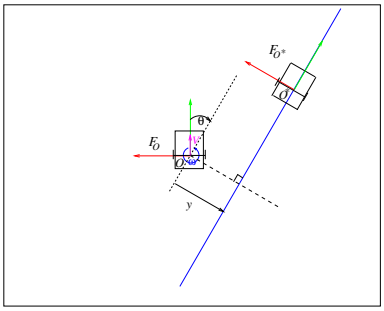


Figure 5: Modelisation of a cart-like following a straight line

3.2 VNHV Control law

Let $\mathbf{X}_v^*$ be the desired state of the VNHV (corresponding to the desired image). The control objective is to reach the straight line Γ on a point with curvilinear coordinate $s < s^*$. To achieve this control objective, chained systems properties are very interesting. Indeed, a chained system results from a conversion of a mobile robot non linear model into an almost linear one, and as long as the robot longitudinal velocity V is non zero, the performances of a path tracking can be determined in terms of settling distance ([11]). In the considered case, such a settling distance can be directly determined in the image space. This property is exploited to ensure that the VNHV reach its desired state. Let us note ω_v the cart angular velocity around the center of its axle. As described previously, the state of the cart in the image plane is given by $\mathbf{X}_v = \begin{bmatrix} s & y & \theta \end{bmatrix}^\top$. The cart control vector is $\mathbf{U} = \begin{bmatrix} V & \omega_c \end{bmatrix}^\top$. These two vectors are related by the following state space model:

$$\begin{cases} \dot{s} = V\cos\theta \\ \dot{y} = V\sin\theta \\ \dot{\theta} = \omega_c \end{cases} \tag{5}$$

Let us now convert the state space model (5) into a chained system of dimension 3 $\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^\top$ with a 2 dimensional control vector $\begin{bmatrix} m_1 & m_2 \end{bmatrix}^\top$:

$$\begin{cases} \dot{a_1} = m_1 \\ \dot{a_2} = a_3 m_1 \\ \dot{a_3} = m_2 \end{cases} \tag{6}$$

The almost linearity of this system is then viewable when the chained system is derived with respect to a_1, with $m_3 = \frac{m_2}{m_1}$:

$$\begin{cases} a_1^{'} = 1 \\ a_2^{'} = a_3 \\ a_3^{'} = m_3 \end{cases} \tag{7}$$

Choosing $a_1 = s$, the linear system (7) is independent from V, since it is driven by a variable homogeneous to the distance covered by the VNHV in the image plane. According to equations (5) and (6), the input variable m_1 is given by $m_1 = \dot{s} = V\cos\theta$. Let us now define $a_2 = y$. Then it comes $\dot{a_2} = V\sin\theta = a_3 m_1$. Therefore a_3 must be chosen as $a_3 = \tan\theta$. Consequently, a_3 is not defined for $\theta = \frac{\pi}{2}\,[\pi]$. However, this condition is generally avoided by choosing an appropriate path to follow. Finally, m_2 is deduced from $m_2 = \dot{a_3}$:

$$m_2 = \cos^2\theta\,(\omega_c - Vc(s)\cos\theta) \tag{8}$$

The final control law has to insure the stabilization of y and θ to zero. A natural and simple choice for the control law is:

$$m_3 = -K_d a_3 - K_p a_2 \;\; (K_p, K_d) \in \mathbb{R}^{+2} \tag{9}$$

This last relation combined to (9) and (7) leads to:

$$a_2^{''} + K_d a_2^{'} + K_p a_2 = 0 \tag{10}$$

Consequently, both a_2 and a_3 converge to zero, independently of the longitudinal velocity of the VHNV as long as $V \neq 0$. In view of $a_2 = y$ and $a_3 = \tan\theta$, (9) ensures that both y and θ converge to zero. Moreover, (10) shows that K_p and K_d determine the performances of a proportional-derivative controller and define a settling distance for the regulation of y. Taking account that V can be negative, the expression of $m_2 = m_3 m_1$ can be obtained from (9) and it is given by:

$$m_2 = -|m_1| K_d a_3 - m_1 K_p a_2 \tag{11}$$

Therefore, ω_c is extracted from (8) in which (11) is inserted:

$$\omega_c(y,\theta) = -V\cos^3\theta K_p y - |V\cos^3\theta| K_d \tan\theta \tag{12}$$

The expression of the control vector $\mathbf{U} = \begin{bmatrix} V & \omega_c \end{bmatrix}^\top$ can thus been computed. The linear velocity V can be considered as a parameter for VNHV path following. As explained previously, this control vector is also a suitable control vector for the real mobile robot. In the case of a cart-like vehicle, since $V \neq 0$ and $\theta \neq \frac{\pi}{2}\,[\pi]$ the equation (12) provides a suitable angular velocity if actuators saturation are not considered. However, V can be seen as a parameter, it can thus be time-varying and, as a consequence, adapted to ensure that ω_c remains between the saturation limits. Moreover, cart-like robots usually navigate in indoor environment with velocities which are far from the saturation limits.

4 Experimental validation

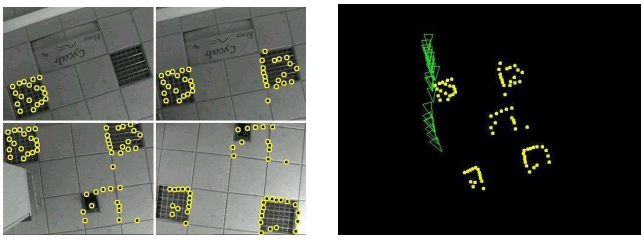
4.1 Environment mapping and localization

The presented approach was tested on a Pekee robot mounted with a low cost camera. Figure 4 shows an example of mosaic construction. The set of images represents the detected 2D models. The last image is a representation of the constructed mosaic and the robot locations (gray triangles) computed using model tracker.

4.2 Testing the vision-based control law

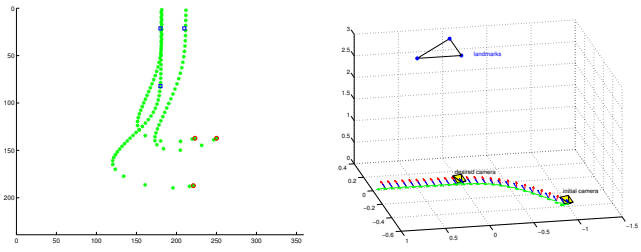
The control law presented in section 3.2 has been tested in a simulation environment. The camera is looking at a three points lying on a plane parallel to the plane of evolution of the mobile robot. Fig. 7 (a) presents the visual features trajectories in image plane.

We first note that the paths described by the features converge to the desired positions. As can be shown in the Figure 7 (b), the corresponding camera path confirms the validity of the control law. As shown in Figure 8 (a) and (b),



(a) (b)

Figure 6: An example of mosaic constructing: (a) a set of images corresponding to detection and modeling of 2D landmarks on the ceiling, (b) Mosaic and robot locations obtained by 2D model tracking



(a) (b)

Figure 7: Visual features and camera trajectories: (a) Visual features trajectories in the image space (Rounds are the initial features, squares are the desired ones), (b) 3D trajectory of the camera

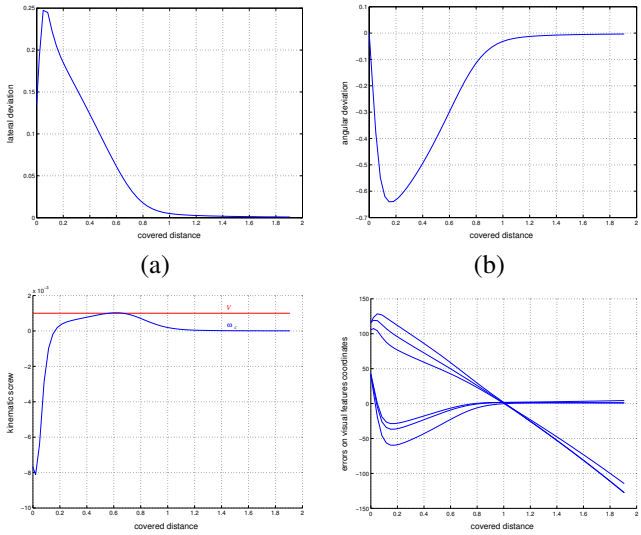
angular and lateral deviations of the VNHV are well regulated to zero. When the straight line is joined, the VNHV stops rotating and the the mobile robot follows straightly the path (Figure 8 (c)). However, we state that the mobile robot runs with a constant value of longitudinal velocity and is not stopped-dead on the desired configuration. This is an inherent issue to the chained systems, which impose $V \neq 0$. That explains why the error in image points coordinates increases after having converged to zero (See Figure 8 (d)). Nevertheless, we do not mind this point when the reached configuration corresponds to a relay image.

5 Conclusion and future works

A method for autonomous indoor environment mapping and image-based navigation has been presented. The approach was developed assuming that the embedded camera looks at a planar surface. Experimental and simulation results confirm the validity of the approach. Future works will be devoted to extend the method to 3D environments.

References

[1] O. Ait Aider, P. Hoppenot, and E. Colle. A model to image straight line matching method for vision-based indoor mobile robot self-location. In *Proc. of the 2002 IEEE/RSJ int. Conference on Itelligent Robots and Systems IROS'2002*, pages 460–465, Lausanne, 2002.



(a) (b) (c) (d)

Figure 8: Signals in the image with respect to the covered distance by the VNHV: (a) lateral deviation (m), (b) angular deviation (rad), (c) mobile robot kinematic screw (V in $m.sec^{-1}$, ω_c in $rad.sec^{-1}$), (d) error in image points coordinates (m)

[2] T. Chateau and J. T. Lapreste. Suivi de nusages de points d'intérêts par filtrage particulaire: application au suivi de vehicules. Technical report, LASMEA, Blaise Pascal University, 2003.

[3] J.B. Hayet. *Contribution à la navigation d'un robot mobile sur amers visuels texturés dans un environnement structuré*. PhD thesis, Université Paul Sabatier, Toulouse, janvier 2003.

[4] J. Kosecka. Visually guided navigation. In *Proc. 4th Int. Symp. on Intelligent Robotic Systems (SIRS'96)*, Lisbon, Portugal, July 1996.

[5] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, ISBN 3-540-76219-1, 1991.

[6] J.P. Laumond. *Robot motion planning and control*, volume 229, chapter Guidelines in nonholonomic motion planning, pages 1–54. Springer, 1998.

[7] Y. Ma, J. Kosecka, and S. S. Sastry. Vision guided navigation for a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, pages 521–37, June 1999.

[8] E. Mouaddib and B. Marhic. Geometrical matching for mobile robot localisation. *IEEE Transactions On Robotics and Automation*, 16(5):542–552, October 2000.

[9] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale invariant features. In *Int. Conference on Robotics and Automation (ICRA'01)*, 2001.

[10] R. Talluri and J.K. Aggarwal. Mobile robot self-location using model-image feature correspondence. *IEEE Trans. on Robotics and Automation*, 12(1):63–77, 1996.

[11] B. Thuilot, C. Cariou, P. Martinet, and M. Berducat. Automatic guidance of a farm tractor relying on a single cp-dgps. *Autonomous Robots*, 13:53–71, 2002.

[12] D. Tsakiris, P. Rives, and C. Samson. Extending visual servoing techniques to nonholonomic mobile robots. In G. Hager D. Kriegman and A. Morse, editors, *The Confluence of Vision and Control*, volume 237 of *LNCIS*, pages 106–117. Springer Verlag, 1998.