Wheeled mobile robots navigation from a visual memory using wide field of view cameras

H. M. Becerra[1], J. Courbon[2], Y. Mezouar[2] and C. Sagues[1]

***Abstract*— In this paper, we propose a visual path following control scheme for wheeled mobile robots based on the epipolar geometry. The control law only requires the position of the epipole computed between the current and target views along the sequence of a visual memory. The proposed approach has two main advantages: explicit pose parameters decomposition is not required and the rotational velocity is smooth or eventually piece-wise constant avoiding discontinuities that generally appear when the target image changes. The translational velocity is adapted as required for the path and the approach is independent of this velocity. Furthermore, our approach is valid for all cameras obeying the unified model, including conventional, central catadioptric and some fisheye cameras. Simulations as well as real-world experiments with a robot illustrate the validity of our approach.**

I. INTRODUCTION

Currently, the development of service robots has attracted the attention of the robotics research community. The locomotion of most of these robots is based on a wheeled platform, and consequently, the strategies to improve their navigation capabilities result of great interest. It is generally accepted that machine vision seems to be a good option of sensory modality for navigation (refer to [1] for a review on visual navigation). This paper describes a new approach of path following based on epipolar geometry and the visual servoing concept [2].

The navigation scheme proposed herein uses the notion of *visual memory*. It means that there is a learning stage in which a set of target images (key images) are stored and they define the path to be replayed in an autonomous stage. This strategy has been introduced for omnidirectional images in [3]. Also a memory–based navigation is proposed in [4] but introducing the prerecorded velocities of the learning stage in the control law. More recently, there are contributions toward the development of autonomous vehicles under this approach. Some of them are position-based approaches, in which, a 3D reconstruction is carried out either using an EKF-based SLAM [5] or a structure from motion algorithm through bundle adjustment [6]. A complete map building is avoided in [7] by relaxing to a local Euclidean reconstruction from the essential matrix using generic cameras. In visual control, image-based approaches generally offer a faster closed loop control with good performance. The work in [8] propose a qualitative visual navigation scheme that is based on some heuristic rules. A Jacobian-based approach that uses the centroid of the abscissas of the feature points is presented in [9]. Most of the mentioned approaches suffer the problem of generating discontinuous rotational velocities when a new key image must be reached. This problem is tackled in [10] for conventional cameras, where the authors propose a varying reference instead of a constant one using the same controller as in [9].

In this paper, we propose a new image-based approach that exploits the epipolar geometry in the context of navigation with a visual memory. Epipolar geometry has been used in visual servoing schemes since it has been introduced in this field by [11]. Some epipolar visual servoing schemes have been proposed for mobile robots with conventional cameras [12], [13]. In these works a total correction of orientation and lateral error is reached while the robot moves always towards the target, unlike [14], where the robot first goes away and then goes back in a second step. This last work has introduced the idea of feedback of the epipoles for central catadioptric cameras. However, none of these approaches are directly extendable to path following because they deal with the pose regulation problem where both input velocities must be computed.

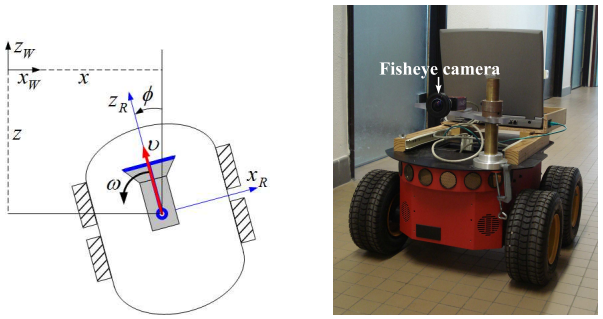
The proposed control scheme uses the value of the current epipole as the only required feedback information. Thus, this approach does not require explicit pose parameters estimation unlike [5], [6]. The visual servoing problem is transformed in a reference tracking problem for the current epipole. It avoids the recurrent problem of discontinuous rotational velocity at key image switching of memory-based schemes that is evident in [7], for instance. The use of epipolar feedback allows us to gather many visual features into a single measurement, which has the advantage of getting a squared control system, where stability of the closed loop can be ensured in contrast to Jacobian-based approaches [9], [10] or heuristic approaches [8]. Additionally, epipolar feedback, as used in our approach, gives the possibility of taking into account valuable a priori information that is available in the visual memory and that is not exploited in previous image-based approaches. We use this information to adapt the translational velocity and also achieve piece-wise constant rotational velocity according to the taught path.

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

This work was supported by project DPI 2009-08126 and grants of Banco Santander-Univ. Zaragoza and Conacyt-México.

[1] H. M. Becerra and C. Sagues are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, C/ María de Luna 1, E-50018 Zaragoza, Spain {hector.becerra, csagues}@unizar.es

[2] J. Courbon and Y. Mezouar are with LASMEA-CNRS-Université Blaise Pascal, 24 Avenue des Landais, 63177 Aubiere, France {jonathan.courbon, youcef.mezouar}@lasmea.univ-bpclermont.fr



(a) (b)

Fig. 1. (a) Robot frame definition, (b) Experimental platform.

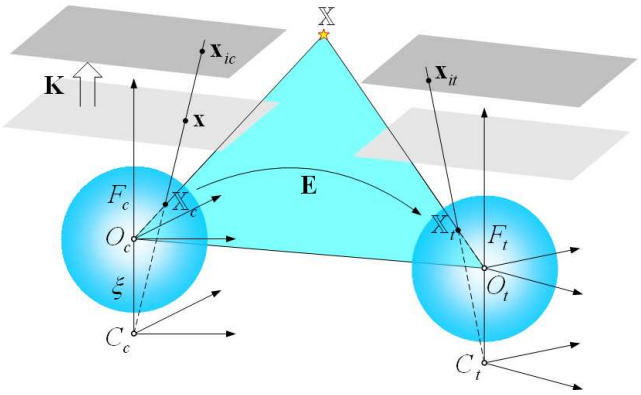


Fig. 2. Generic model of the image formation and epipolar geometry between generic central cameras.

At this aim, the generic camera model introduced in [15] is exploited to design our control strategy. This means that the proposed method can be applied not only to conventional cameras but also to all central catadioptric cameras and to a large class of fisheye cameras [16], since the epipolar geometry can be computed from the unified model when the camera parameters are known (the calibration can be estimated for instance using the tools described in [17]).

The paper is organized as follows. Section II introduces the robot and camera model, and the epipolar geometry obeying such model. Section III details the proposed control strategy. Section IV presents the performance of the visual navigation via simulations and real world experiments and finally, Section V provides the conclusions.

II. MATHEMATICAL MODELING

A. Robot Kinematics

Let $\chi = (x, z, \phi)^T$ be the state vector of a differential drive robot (Fig. 1), where $x(t)$ and $z(t)$ are the robot position coordinates in the plane, and $\phi(t)$ is the robot orientation. The kinematic model of the robot expressed in state space can be written as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\sin\phi & 0 \\ \cos\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

being $v(t)$ and $\omega(t)$ the translational and angular input velocities, respectively.

B. Epipolar Geometry for Generic Cameras

The constrained field of view of conventional cameras can be enhanced using wide field of view imaging systems such as fisheye cameras or full view omnidirectional cameras. It is known that the imaging process performed by conventional and catadioptric cameras can be modeled by a unique representation [15]. Such unified projection model works properly for imaging systems having a single center of projection. Although fisheye cameras do not accomplish such property, some recent experimental results have shown that the unified projection model is able to represent their image formation process with the required accuracy for robotic applications [16].

The unified projection model describes the image formation as a composition of two central projections. The first is a central projection of a 3D point onto a virtual unitary sphere and the second is a perspective projection onto the image plane. In this work we assume that the camera is calibrated [17], which allows us to exploit the representation of the points on the unit sphere. Let denote a 3D point as $\mathbb{X}$, and its corresponding coordinates as $\mathbf{X}$. Thus, point coordinates on the sphere $\mathbf{X}_c$ can be computed from point coordinates on the normalized image plane $\mathbf{x}$ (refer to Fig. 2) and the sensor parameter ξ as follows

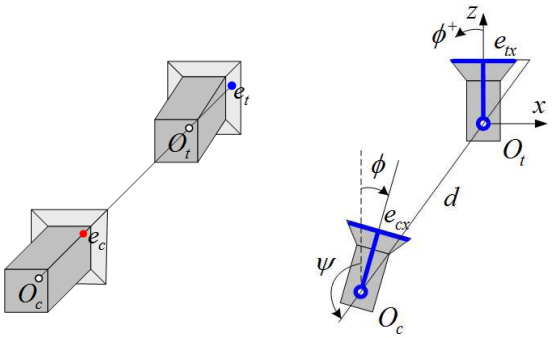
$$\begin{aligned} \mathbf{X}_c &= \left(\eta^{-1} + \xi\right) \bar{\mathbf{x}}, \\ \bar{\mathbf{x}} &= \begin{bmatrix} \mathbf{x}^T & \frac{1}{1+\xi\eta} \end{bmatrix}^T, \end{aligned} \tag{2}$$

where $\eta = \frac{-\gamma - \xi\left(x^2+y^2\right)}{\xi^2\left(x^2+y^2\right)-1}$, $\gamma = \sqrt{1 + \left(1-\xi^2\right)\left(x^2+y^2\right)}$.

Regarding to Fig. 2, let $\mathbb{X}$ be a 3D point and let $\mathbf{X}_c$ and $\mathbf{X}_t$ be the coordinates of that point projected onto the unit spheres of the current F_c and target frame F_t. The epipolar plane contains the effective viewpoints of the imaging systems O_c and O_t, the 3D point $\mathbb{X}$ and the points $\mathbb{X}_c$ and $\mathbb{X}_t$. The coplanarity of these points leads to the well known epipolar constraint

$$\mathbf{X}_c^T \; \mathbf{E} \; \mathbf{X}_t = 0, \tag{3}$$

being $\mathbf{E}$ the essential matrix relating the pair of normalized virtual cameras. Normalized means that the effect of the known calibration matrix has been removed. As typical, from this geometry it is possible to compute the epipoles as the points lying on the base line and intersecting the corresponding virtual image plane. In order to avoid singularities in the epipolar geometry three views can be used to estimate this geometry as proposed in [18]. Fig. 3(a) shows the epipoles of a configuration of the pair of virtual cameras with center of projection O_c and O_t respectively. Fig. 3(b) presents an upper view of this configuration, where the framework of the epipolar geometry constrained to planar motion is defined. A global reference frame centered in the origin $O_t = (0, 0, 0)$ of the target viewpoint is defined. Then, the current camera location with respect to this reference is $O_c = (x, z, \phi)$. Assuming the described framework in Fig. 1,



(a) (b)

Fig. 3. (a) 3D epipolar geometry, (b) Planar epipolar geometry framework.

where the camera location coincides with the robot location, the epipoles can be written as a function of the robot state as follows

$$\begin{aligned} e_{cx} &= \alpha_x \frac{x\cos\phi + z\sin\phi}{z\cos\phi - x\sin\phi}, \\ e_{tx} &= \alpha_x \frac{x}{z}. \end{aligned} \tag{4}$$

Cartesian coordinates x and z can be expressed as a function of the polar coordinates d and ψ as

$$x = -d\sin\psi, \quad z = d\cos\psi, \tag{5}$$

with $\psi = -\arctan(e_{tx}/\alpha_x)$, $\phi - \psi = \arctan(e_{cx}/\alpha_x)$ and $d^2 = x^2 + z^2$. For the case of normalized cameras $\alpha_x = 1$ in (4) and in the subsequent equations.

III. NAVIGATION STRATEGY

There are some works that use the epipoles as direct feedback in the control law for a pose regulation task [12], [13], [14]. In the first two works the robot moves directly toward the target, but the translational velocity computation suffers of singularity problems, which make non-feasible its direct application for navigation. In the last work, the effort to avoid the singularity takes the robot to perform some inappropriate maneuvers for navigation. We propose to use only the x-coordinate of the current epipole as feedback information to modify the robot heading and so, to correct the lateral deviation. The current epipole gives information of the translation direction and it is directly related to the required robot rotation to be aligned with the target. As can be seen in Fig. 4, $e_{cx} = 0$ means that the longitudinal camera axis of the robot is aligned with the baseline and the camera is looking directly toward the target. Therefore, the control goal is to take this epipole to zero in a smooth way, which is achieved by using an appropriate reference. It allows avoiding discontinuous rotational velocity when a new target image is required to be reached. Additionally, we propose to take into account some a priori information of the shape of the visual path that can be obtained from the epipoles relating two consecutive key images. This allows us to adapt the translational velocity and also achieve piece-wise constant rotational velocity according to the taught path.

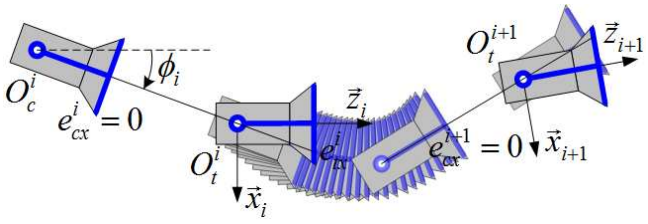


Fig. 4. Control strategy based on zeroing the current epipole.

A. Visual Memory Building

Although the scope of the paper is the control strategy, we briefly outline the procedure to build a visual memory. The visual memory defines a path to be replayed in the autonomous navigation stage. A sequence of images are stored from the onboard camera during a learning stage by manual driving of the robot. We assume that during learning, the translational velocity is never zero. From all the captured images a reduced set is selected as key images by ensuring a minimum number of shared features between two consecutive key images. For more details about the memory building refer to [7]. We assume that n key images are chosen and that these images are separated along the path by a minimum distance $d_{\min}$.

B. Control Law for Autonomous Navigation

Let us define a unidimensional task function to be zeroed that depends on the current epipole e_{cx}. This allows us to gather many visual features into a single measurement, which has the benefit of getting a squared control system. So, potential stability problems are avoided unlike previous Jacobian approaches [9], [10]. In the sequel, we avoid the use of the subscript x. This function represents the tracking error of the current epipole e_c with respect to a desired reference $e_c^d(t)$

$$\zeta_c = e_c - e_c^d(t). \tag{6}$$

The tracking error is defined using the i^{th} key image as target, although it is not indicated explicitly. The following nonlinear differential equation represents the rate of change of the tracking error as given by both input velocities and is obtained by taking the time-derivative of (6) and using the polar coordinates (5)

$$\dot{\zeta}_c = -\frac{\alpha_x \sin(\phi - \psi)}{d\cos^2(\phi - \psi)}\upsilon + \frac{\alpha_x}{\cos^2(\phi - \psi)}\omega_t - \dot{e}_c^d. \tag{7}$$

The subscript of the rotational velocity ω_t refers to the velocity for reference tracking. We define the desired behavior through the following differentiable sinusoidal reference

$$\begin{aligned} e_c^d(t) &= \frac{e_c(0)}{2}\left(1 + \cos\left(\frac{\pi}{\tau}t\right)\right), \ 0 \le t \le \tau \\ e_c^d(t) &= 0, \qquad\qquad\qquad\quad t > \tau \end{aligned} \tag{8}$$

where $e_c(0)$ is the value of the current epipole at the beginning or at the time of key image switching and τ is

a suitable time in which the current epipole must reach zero, before the next switching of key image. Thus, a timer is restarted at each instant when a change of key image occurs. The time required in the reference can be easily replaced by a number of iterations in the control cycle. Note that this reference trajectory provides a smooth zeroing of the current epipole from its initial value. Let us express the equation (7) as follows

$$\dot{\zeta}_c = \mu_\upsilon + \frac{\alpha_x}{\cos^2(\phi-\psi)}\omega_t - \dot{e}_c^d, \tag{9}$$

where $\mu_\upsilon = -\frac{\alpha_x \sin(\phi-\psi)}{d\cos^2(\phi-\psi)}\upsilon$ represents a known disturbance depending on the translational velocity. The velocity ω_t can be found by using Input-Output Linearization of the error dynamics. Thus, the following rotational velocity assigns a new dynamics through the auxiliary input δ_a

$$\omega_t = \frac{\cos^2(\phi-\psi)}{\alpha_x}\left(-\mu_\upsilon + \dot{e}_c^d + \delta_a\right).$$

We define the auxiliary input as $\delta_a = -k_{c_i}\zeta_c$ to keep the current epipole tracking the reference trajectory, where $k_c > 0$ is a control gain. Thus, the resulting rotational velocity is

$$\omega_t = \frac{\sin(\phi-\psi)}{d}\upsilon + \frac{\cos^2(\phi-\psi)}{\alpha_x}\left(\dot{e}_c^d - k_c\zeta_c\right). \tag{10}$$

This velocity reduces the error dynamics to $\dot{\zeta}_c = -k_c\zeta_c$. So, the tracking error exhibits an exponentially stable behavior, with settling time $\gamma \approx 5/k_c$. Since that the control goal of this controller is the tracking, ω_t starts and finishes at zero for every key image. In order to maintain the velocity around a constant value we propose to add a term for a nominal rotational velocity $\bar{\omega}$. The next section describes how this nominal velocity is obtained. So, the rotational velocity can be eventually computed as

$$\omega = k_t\omega_t + \bar{\omega}, \tag{11}$$

where $k_t > 0$ is a weighting factor on the reference tracking control ω_t. It is worth emphasizing that the velocity ω_t by itself is able to drive the robot along the path described by the image memory, however, the total input velocity in (11) behaves more natural around constant values. We will refer to the only reference tracking control, ω_t (10), as RT and the complete control, ω (11), as RT+.

C. Exploiting Information from the Memory

All previous image-based approaches for navigation using a visual memory only exploit local information, i.e., the required rotational velocity is only computed from the current and the next nearest target images. We propose to exploit the visual memory in order to have an a priori information about the whole path without the need of a 3D reconstruction or representation of the path, unlike [5], [6], [7]. A kind of qualitative map of the path can be easily obtained from the current epipole relating two consecutive key images of the memory, which is denoted by e_c^m. Thus, $e_{c_i}^m$ shows qualitatively the orientation of the camera in the $(i-1)^{th}$ key image with respect to the i^{th} one and so, it gives an idea of the curvature of the path.

We propose to use this a priori information to apply an adequate translational velocity and to compute the nominal rotational velocity that appears in (11). As before, we suppress the subscript i, but recall that the epipole e_c^m is computed between all consecutive pairings of key images. The translational velocity is changed smoothly for every switching of key images using the following mapping $e_c^m \rightarrow (\upsilon_{\min}, \upsilon_{\max})$

$$\upsilon = \upsilon_{\max} + \upsilon_{\min} + \frac{\upsilon_{\max}-\upsilon_{\min}}{2}\tanh\left(1 - \frac{|e_c^m/d_{\min}|}{\sigma}\right) \tag{12}$$

where σ is a positive parameter that determines the distribution of the velocities. Once a translational velocity is set from the previous equation for each key image, υ can be used to compute the nominal velocity $\bar{\omega}$ as follows ($\bar{\omega} \propto e_c^m$)

$$\bar{\omega} = \frac{k_m\upsilon}{d_{\min}}e_c^m \tag{13}$$

where $k_m < 0$ is a constant factor to be tuned. This velocity by itself is able to drive the robot along the path, but correction is introduced in (11) through (10).

D. Timing Strategy and Key Image Switching

It is clear that there is a need to zero the current epipole before reaching the next key image during the navigation, which imposes a constraint for the time τ. Thus, a strategy to define this time is related to the minimum distance between key images ($d_{\min}$) and the translational velocity (υ) for each key image as follows:

$$\tau = \frac{d_{\min}}{\upsilon}.$$

We have found that a good approach to relate this time with the settling time γ of the tracking error is to consider $0.4\tau = 5/k_c$, from which $k_c = 12.5/\tau$.

By running the controller (9) with the reference (8), the time τ and the control gain k_c as described above, an intermediate location determined by $d_{\min}$ is reached. In the best case, when $d_{\min}$ coincides with the real distance between key images, the robot reaches the location of the corresponding key image. In order to achieve a good correction of the longitudinal position for each key image, the reference (8) is maintained to zero, which implies that $\omega = 0$, until the image error starts to increase. The *image error* is defined as the mean squared error between the r corresponding image points of the current image ($P_{i,j}$) and points of the next closest target key image (P_j), i.e.,

$$\epsilon = \frac{1}{r}\sum_{j=1}^{r}\|P_j - P_{i,j}\| \tag{14}$$

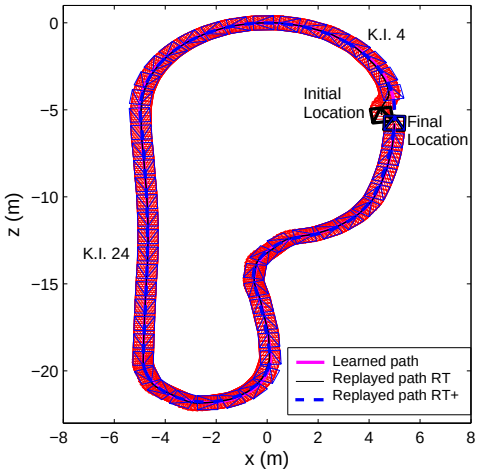
As shown in [8], the image error decrease monotonically until the robot reaches each target view. In our case, the increment of the image error is the switching condition for the next key image, which is confirmed by using the current and the previous difference of instantaneous values of the image error.

IV. EXPERIMENTAL EVALUATION

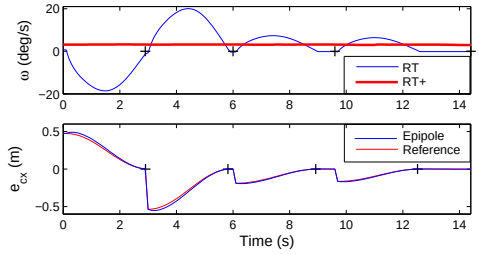
A. Simulations Results

In this section, we present some simulations in Matlab of our navigation scheme. We use the generic camera model [15] to generate synthetic key images from a 3D scene according to the robot motion on a predefined path. This learned path starts in the location (5,-5,0^o) and finishes just before to close the loop of 54 m long. The camera parameters are $\alpha_x = 222.9$, $\alpha_y = 222.1$, $x_0 = 305.1$, $y_0 = 266.9$ all of them in pixels, $\xi = 2.875$ and the size of the images is 640×480 pixels. These camera parameters are also used to compute the points on the sphere (2) from the image coordinates. In these simulations, a typical 8-points algorithm has been used to estimate the essential matrix [19]. The only required feedback information (e_{cx}) is computed as the right null space of the essential matrix $\mathbf{E}\,[e_{cx}, e_{cy}, e_{cz}]^T = 0$.

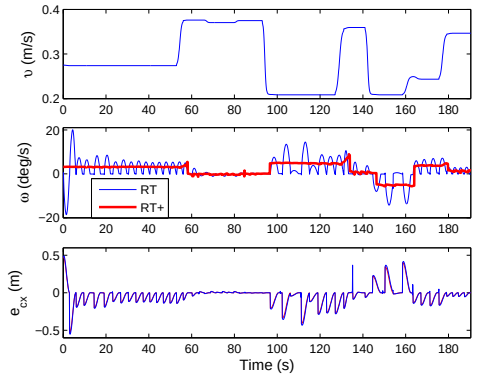
The first simulation uses a fix distance between key images of one meter, i.e., there are 54 key images. The translational velocity is bounded between 0.2 m/s and 0.4 m/s. In order to set the time τ and the control gain k_c, it is assumed a minimum distance between key images $d_{\min} = 0.8$ m. We present the results for two cases according to (11): 1) only reference tracking (RT) and 2) reference tracking + nominal velocity (RT+). The applicability of the last is limited to start on the path and the former is able to correct an initial position out of the path. We can see in Fig. 5(a) that the resultant path of the autonomous navigation stage is almost similar to the learned one in both cases. Although the initial location is out of the learned path for the RT, the robot achieves the tracking just in the second key image. The first plot of Fig. 5(b) shows the behavior of the rotational velocity for the four first key images. On one hand, we can see that this velocity is smooth for the RT case. The velocity starts to grow always from zero in the marked points, which correspond to changes of key image, and returns to zero at the next switching. On the other hand, we have a constant velocity for the RT+. The third plot of the same figure presents the reference tracking of the epipole for the RT with a mark when it reaches zero. Fig. 5(c) presents the varying translational velocity as given by (12) for the whole path. The evolution of this velocity agrees with the level of curvature of the path. Fig. 5(c) shows the evolution of the rotational velocity and the reference tracking for the epipole along the whole path. The addition of the nominal value allows to achieve a piece-wise constant rotational velocity.



(a) Resultant path and key images distribution.



(b) Rotational velocity and epipole for the first 4 key images.

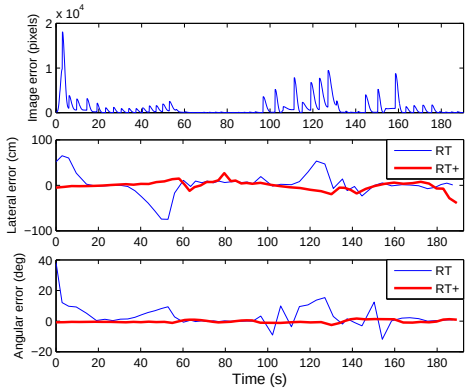


(c) Velocities and epipole evolution for the whole path.

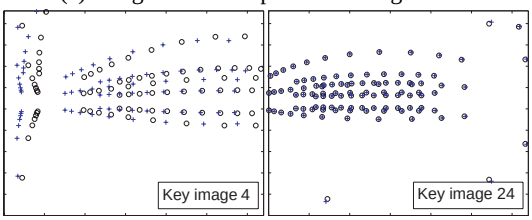
Fig. 5. Simulation results using a fix distance between key images (1 m), for both cases: only reference tracking (RT) and adding the nominal rotational velocity (RT+). In both cases the varying translational velocity is the same.

Fig. 6 presents the performance of the approach for the same experiment. The first plot of Fig. 6(a) shows the behavior of the image error for the RT case. During the first seconds, the error increases because the robot is out of the path. In the subsequent steps, from the second key image, this error exhibits a monotonic decay at each step. After that, the largest peaks in the image error correspond to the sharp curves in the path, which also causes the highest error in the path following. We can see in the plots of the errors to reach each key image in the same figure that the RT+ control obtains best tracking performance than the RT control for this condition of fixed distance between key images. The snapshots of Fig. 6(b) show that the points features of key images are reached with good precision even in curves.

The performance of our navigation scheme including image noise has been also evaluated. In this case, 35 key images are placed randomly along the predefined path, in such a way that a minimum distance $d_{\min} = 1.40$ m is assumed. A Gaussian noise with standard deviation of 0.5 pixels is added to the image coordinates. The path following is still good along the whole path for the RT control (Fig. 7(a)) and adequate for the RT+. The RT+ control is slightly



(a) Image error and path following errors.



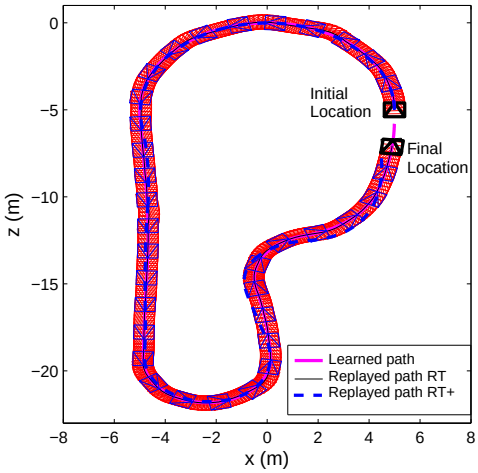
(b) Snapshots reaching key images (current: +, target: O).

Fig. 6. Performance of the navigation task for the results in Fig. 5.

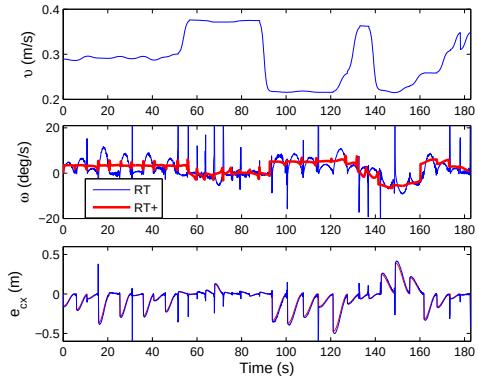
sensitive to longer and random distance between key images along sharp curves. The RT performs well in spite of that the current epipole and the rotational velocity are noisy (Fig. 7(b)). However, the rotational velocity as obtained from the RT+ result more convenient for a real application. The path errors to reach each key image are comparable for both controllers, as can be seen in Fig. 7(c).

B. Real-world Experiments

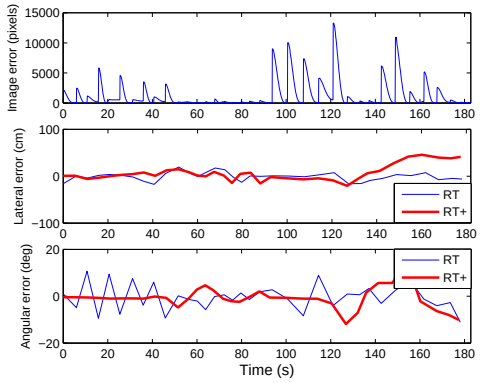
In order to test the proposed control law we have used the software platform described in [7]. This software selects a set of key images to be reach from a sequence of images that is acquired in a learning stage. It also extracts features from the current view and the next closest key image, matches the features between these two images at each iteration and computes the current epipole that relates the two views. The interest points are detected in each image with Harris corner detector and then matched by using a Zero Normalized Cross Correlation score. This method is almost invariant to illumination changes and its computational cost is small. The software is implemented in C++ and runs on a common laptop. Real world experiments have been carried out for indoor navigation along a living room with a Pioneer robot (Fig. 1(b)). The imaging system consist of a Fujinon fisheye lens and a Marlin F131B camera looking forward, which provides a field of view of 185 deg. The size of the images is 640×480 pixels. A constant translational velocity $\upsilon = 0.1$ m/s is used and a minimum distance between key images $d_{\min} = 0.6$ is assumed. Fig. 8(a) shows the resultant and learned paths for one of the experimental runs as given by the robot odometry. In this experiment, we test the RT control since the initial robot position is out of the learned path. We can see that after some time, the reference path is reached



(a) Resultant path and key images distribution.



(b) Velocities and current epipole evolution.

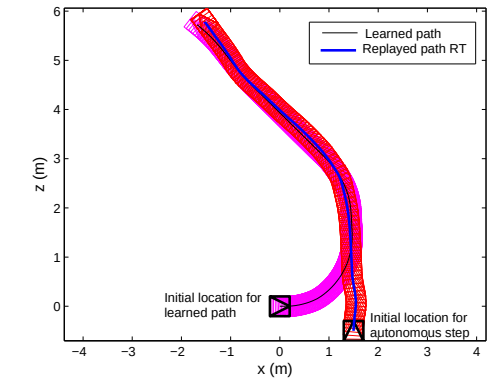


(c) Image error and path following errors.

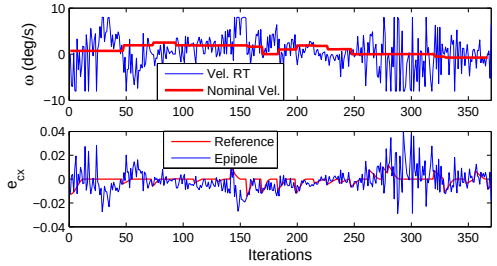
Fig. 7. Simulation including image noise ($\sigma = 0.5$ pixels), random distance between key images (from 1.45 to 1.6 m) and varying translational velocity between $0.2 - 0.4$ m/s.

and followed closely. The computed rotational velocity and the behavior of the current epipole are presented in Fig. 8(b). The robot follows the visual path until a point where there is not enough number of matched features. In the same figure, we depict the nominal rotational velocity as computed offline only to show that it agrees the shape of the path. In Fig. 8(c) we can see that the image error is not reduced initially

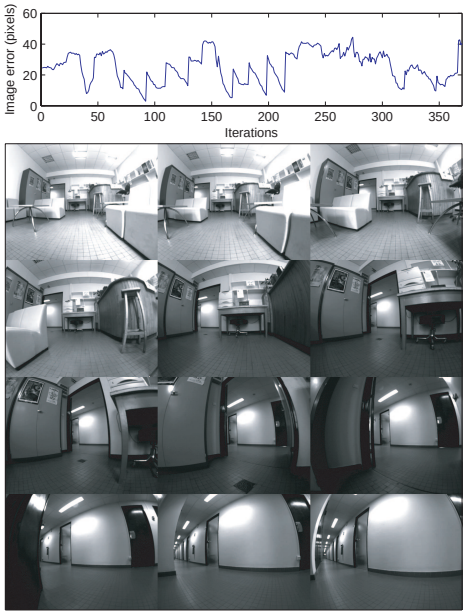
because the robot is out of the path, but after it is reached, the image error for each key image is reduced. The same figure presents a sequence of images as acquired for the robot camera during the navigation.



(a) Learned path and resultant replayed path.



(b) Rotational velocity and current epipole evolution.



(c) Image error and sequence of images during navigation.

Fig. 8. Real world experiment for indoor navigation with a fish eye camera.

V. CONCLUSIONS

Along this paper, we have proposed a control scheme for wheeled mobile robots that uses the epipolar geometry to compute velocities in order to follow a visual path. The value of the current epipole is the unique required information for the control law. This is an image-based approach because no explicit pose parameters decomposition is carried out. The scheme avoids discontinuous rotational velocity when a new target image must be reached. Eventually, the this velocity can be piece-wise constant. The translational velocity is adapted according to the path and the approach is independent of its value. We exploit the advantages of wide field of view cameras, in particular fisheye. The camera calibration parameters are required for the epipolar geometry of this kind of cameras and can be easily obtained with the available calibration tools. The proposed scheme has presented a good performance according to the simulation results and real world experiments.

REFERENCES

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

[2] F. Chaumette and S. Hutchinson. Visual servo control part I: Basic approaches. *IEEE Robotics and Autom. Mag.*, 13(14):82–90, 2006.

[3] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue. Visual navigation using using omnidirectional view sequence. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 317–322, 1999.

[4] D. Burschka and G. Hager. Vision-based control of mobile robots. In *Int. Conf. on Robotics and Automation*, pages 1707–1713, 2001.

[5] T. Goedeme, T. Tuytelaars, L. V. Gool, G. Vanacker, and M. Nuttin. Feature based omnidirectional sparse visual path following. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1806–1811, 2005.

[6] E. Royer, M. Lhuillier, M. Dhome, and J. M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3):237–260, 2007.

[7] J. Courbon, Y. Mezouar, and P. Martinet. Autonomous navigation of vehicles from a visual memory using a generic camera model. *IEEE Trans. on Intel. Transportation Systems*, 10(3):392–402, 2009.

[8] Z. Chen and S. T. Birchfield. Qualitative vision-based mobile robot navigation. In *IEEE Int. Conf. on Robotics and Automation*, pages 2686–2692, 2006.

[9] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette. Outdoor visual path following experiments. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 4265–4270, 2007.

[10] A. Cherubuni and F. Chaumette. Visual navigation with a time-independent varying reference. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 5968–5973, 2009.

[11] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. *Int. Journal of Computer Vision*, 33(2):117–137, 1999.

[12] G. López-Nicolás, C. Sagüés, J.J. Guerrero, D. Kragic, and P. Jensfelt. Nonholonomic epipolar visual servoing. In *IEEE Int. Conf. on Robotics and Automation*, pages 2378–2384, 2006.

[13] H. M. Becerra and C. Sagues. A sliding mode control law for epipolar visual servoing of differential-drive robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3058–3063, 2008.

[14] G. L. Mariottini, D. Prattichizzo, and G. Oriolo. Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera. In *IEEE Int. Conf. on Robotics and Automation*, pages 538–544, 2006.

[15] C. Geyer and K. Daniilidis. An unifying theory for central panoramic systems and practical implications. In *European Conf. on Computer Vision*, pages 445–461, 2000.

[16] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A generic fisheye camera model for robotics applications. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1683–1688, 2007.

[17] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE Int. Conf. on Robotics and Automation*, pages 3945–3950, 2007.

[18] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. on Pattern Anal. and Mach. Intel.*, 26(6):756–770, 2004.

[19] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, second edition, 2004.