

Visual navigation of wheeled mobile robots using direct feedback of a geometric constraint

Héctor M. Becerra¹, Carlos Sagüés², Youcef Mezouar³, and Jean-Bernard Hayet ^{*1}

¹Centro de Investigación en Matemáticas (CIMAT), C.P. 36240, Guanajuato, Gto., Mexico. E-mails: hector.becerra@cimat.mx, jbhayet@cimat.mx

²Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, C/ María de Luna 1, E-50018, Zaragoza, Spain. E-mail: csagues@unizar.es

³Institut Français de Mécanique Avancée, Institut Pascal, 63177, Aubière, France. E-mail: youcef.mezouar@ifma.fr

Abstract

Many applications of wheeled mobile robots demand a good solution for the autonomous mobility problem, i.e., the navigation with large displacement. A promising approach to solve this problem is the following of a visual path extracted from a visual memory. In this paper, we propose an image-based control scheme for driving wheeled mobile robots along visual paths. Our approach is based on the feedback of information given by geometric constraints: the epipolar geometry or the trifocal tensor. The proposed control law only requires one measurement easily computed from the image data through the geometric constraint. 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 in previous works when the target image changes. The translational velocity is adapted as demanded for the path and the resultant motion is independent of this velocity. Furthermore, our approach is valid for all cameras with approximated central projection, including conventional, catadioptric and some fisheye cameras. Simulations and real-world experiments illustrate the validity of the proposal.

Key words: Visual navigation, visual path following, visual memory, epipolar geometry, trifocal tensor

1 Introduction

The strategies to improve the navigation capabilities of wheeled platforms result of great interest in robotics and particularly in the field of service robots. A good strategy for visual navigation is based on the use of the so-called *visual*

memory. This approach consists of two stages. First, there is a learning stage in which a set of images is stored to represent the environment. Then, a subset of those images (key images) is selected to define a path to be followed in an autonomous stage. This approach may be applied for autonomous personal transportation in places under structured demand, like airport terminals, attraction resorts or university campuses, etc. The visual memory approach has been introduced in [1] for conventional cameras and extended in [2] for omnidirectional cameras. Later, some position-based schemes relying on the visual memory approach have been proposed with a 3D reconstruction carried out either using an EKF-based SLAM [3], or a structure from motion algorithm through bundle adjustment [4]. A complete map building is avoided in [5] and [6] by relaxing to a local Euclidean reconstruction from the homography and essential matrix decomposition respectively, using generic cameras.

In general, image-based schemes for visual path following offer good performance with higher closed loop frequency. The work in [7] proposes 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 [8]. Most of the aforementioned approaches suffer the problem of generating discontinuous rotational velocities when a new key image must be reached. This problem is tackled in [9] for conventional cameras, where the authors propose the use of a time-independent varying reference. A different approach of visual path following is presented in [10], where the path is defined by a line painted on the floor.

The approach based on a visual memory is a mature paradigm for robot navigation that allows a path to be replayed at different conditions than in the learning phase. Even more important, this approach has the benefits of the closed loop control, in contrast to a simplistic approach of replying

*This work was supported by projects DPI 2009-08126 and DPI 2012-32100 and grants of Banco Santander-Universidad de Zaragoza and Conacyt-México.

the control commands of the training phase in open loop. It is worth noting that, as opposed to the trajectory tracking problem, in the path following problem there is not a time-based parametrization of the path. Additionally, in the path following problem we are not interested in reaching a strict desired configuration as in the pose regulation problem [11]. The visual path following problem with collision avoidance has been tackled in [12] relying on an on-board range scanner for obstacle detection, given the complexity of this task using a pure vision system.

In this paper, we propose an image-based scheme that exploits the direct feedback of a geometric constraint in the context of navigation with a visual memory. The proposed control scheme uses the feedback of only one measurement, the value of the current epipole or one element of the trifocal tensor (TT). The use of a geometric constraint allows us to gather many visual features into a single measurement. The scheme exploiting the epipolar geometry (EG) has been introduced in a preliminary paper [13] focusing on the use of fisheye cameras. In the work herein, in addition to the extension of using the TT as visual measurement, a comprehensive comparison of different control proposals is carried out using omnidirectional vision. The proposed approach does not need explicit pose parameters estimation unlike [3], [4]. The visual control problem is transformed in a reference tracking problem for the corresponding measurement. The reference tracking avoids the recurrent problem of discontinuous rotational velocity at key image switching of memory-based schemes that is revealed in [5], [14] and [6], for instance.

We contribute to the problem of visual path following by using direct feedback of a geometric constraint (EG or TT). Feedback of a geometric constraint has been previously used in the context of the pose regulation problem, e.g., [15], [16], [17]; however, none of the existing works can be directly extended to the navigation problem. The path following problem essentially requires the computation of the rotational velocity, and so, the use of one measurement provides the advantage of getting a squared control system, where stability of the closed loop can be ensured similarly to the Jacobian-based schemes [8], [18] and in contrast to heuristic schemes [7]. Although a squared control system is not indispensable for applying a Jacobian based scheme, it is advantageous. The geometric constraints, as used in our approach, give the possibility of taking into account valuable a priori information that is available in the visual path and that is not exploited in previous image-based approaches. This information, used as feedforward control, permits achieving piece-wise constant rotational velocity according to the learned path without discontinuities when a new reference image is given. Additionally, this a priori information is used to adapt the translational velocity according to the shape of the path.

Conventional cameras suffer from a restricted field of view.

Many approaches in vision-based robot control, such as the one proposed in this paper can benefit from the wide field of view provided by omnidirectional or fisheye cameras. We also contribute exploiting the generic camera model [19] to obtain a generic control scheme. This means that the proposed approach can be applied not only for conventional cameras but also for all central catadioptric vision systems and some fisheye cameras.

In summary, the contributions of the paper are the following:

1. The proposal of two task functions based on a geometric constraint, the EG and the TT, enabling to have a direct control of the rotational velocity of the robot to achieve an accurate visual path following, letting the translational velocity as an independent parameter.
2. Two different control schemes are developed to exploit the proposed task functions, one is a pure feedback control and the other includes a strong component of feedforward control. Both control schemes avoid discontinuities when a new reference image is given, however, the scheme using a feedforward term computes a preferred piece-wise constant rotational velocity.
3. A comprehensive comparative study between the different control proposals is reported, which shows that the TT based control gives better performance for general setups in terms of smoothness of robot velocity and robustness to noise.
4. The proposed navigation scheme is valid for a large class of vision systems: conventional perspective cameras, central catadioptric systems (e.g. hypercatadioptric, paracatadioptric) and some fisheye cameras.

The paper is organized as follows. Section 2 outlines the visual memory approach and presents the overall scheme addressed in this work. Section 3 presents the mathematical modeling for the wheeled mobile robot and the computation of the visual measurements obtained from generic cameras. Section 4 describes the proposed navigation strategies on the basis of the information provided by the EG or the TT. In Section 5, the control law for autonomous navigation for both geometric constraints is detailed. Section 6 shows the performance of the control scheme via simulations and real-world experiments, and finally, Section 7 summarizes the conclusions.

2 The visual memory approach

The framework for navigation based on a visual memory consists of two stages. The first one is a learning stage where the

visual memory is built. In this stage, the user guides the robot along the places where it is allowed to move. A sequence of images is stored from the onboard camera during this stage in order to represent the environment. From all the captured images, a reduced set is selected as key images. A minimum number of visual features must exist between two consecutive key images. After that stage, the robot is requested to reach a specific location in the environment defined by a target image. Initially, the robot has to localize itself by comparing the information that it is currently seeing with the set of key images. The robot is localized when the “more similar” key image to the current image is found. Then, a visual path that connects the current location with the target location as a sequence of n key images is generated. This path should be followed in the autonomous navigation stage in order to reach the target location. Although no metric information will be used in our approach, we assume that there is at least a distance d_{\min} between robot locations associated to any two consecutive key images. This parameter will be used in Sections 5.1-5.2 for a feedforward control and for a time-based strategy of key image switching. This assumption implies that the robot could stop during the learning stage, however, a key image is not stored until the robot continues defining the path and the number of visual features is less than a threshold.

Figure 1 presents an overview of the proposed framework for visual path following. Given the visual path with n key images, we assume that an image-based localization component provides the first key image to be reached for the starting of the autonomous navigation stage (e.g., I_i). Then, image point features are matched between the current image of the onboard camera and the corresponding key image. The matched point features are used: 1) to compute a geometric constraint that is the basis of the proposed control law and, 2) to compute an image error that gives the switching condition to the next key image. When the image error is small, a new key image is requested to be reached and the same cycle is repeated until the final key image I_n is reached. Our interest is to have a target switching criteria that is valid for different kind of control laws, while limiting the extraction of information from image points coordinates. The use of the same geometric constraint for target switching might need a particular strategy for each case or might require a partial Euclidean reconstruction as addressed in [5], [6].

It is worth noting that we focus on the design of a control scheme for the autonomous navigation stage. In this sense, we assume that the visual path has been generated appropriately. The visual path is provided by a higher level layer that deals with aspects like the selection of key images, the topological organization of the key images, the planning of a visual path and the initial localization. For more details about these aspects refer to [6] and [8]. For features detection and matching in the context of memory-based visual navigation refers to [3]

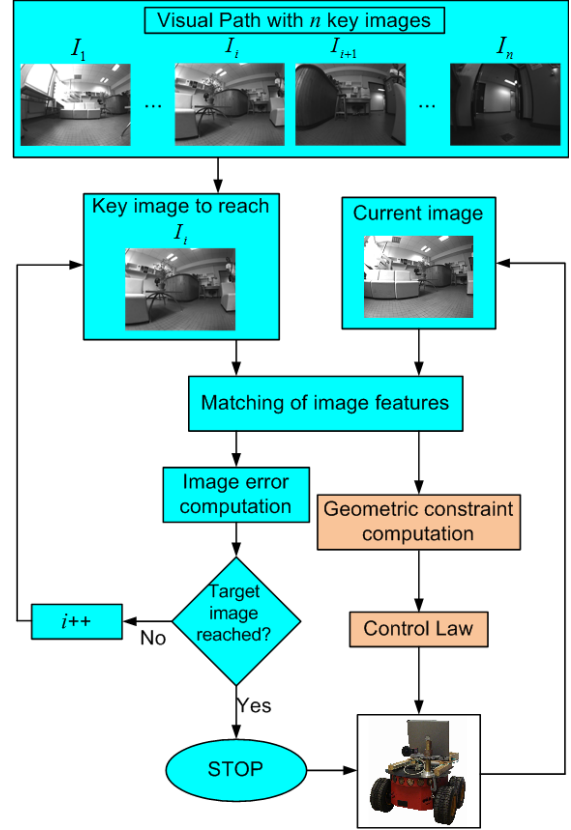


Figure 1: General scheme of the navigation based on the visual memory approach.

and [6].

3 Robot model and visual measurements

In this section, we introduce the required mathematical modeling in order to relate the kinematic motion of the robot to the visual measurements. As we are interested in that the navigation scheme can be valid for any central vision system, the generic camera model of [19] is summarized, as well as the method to estimate the geometric constraints that are the basis of the proposed control law.

3.1 Robot kinematics

Let $\chi = (x, y, \phi)^T$ be the state vector of a differential-drive robot under the frame depicted in figure 2(a), where $x(t)$ and $y(t)$ 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:

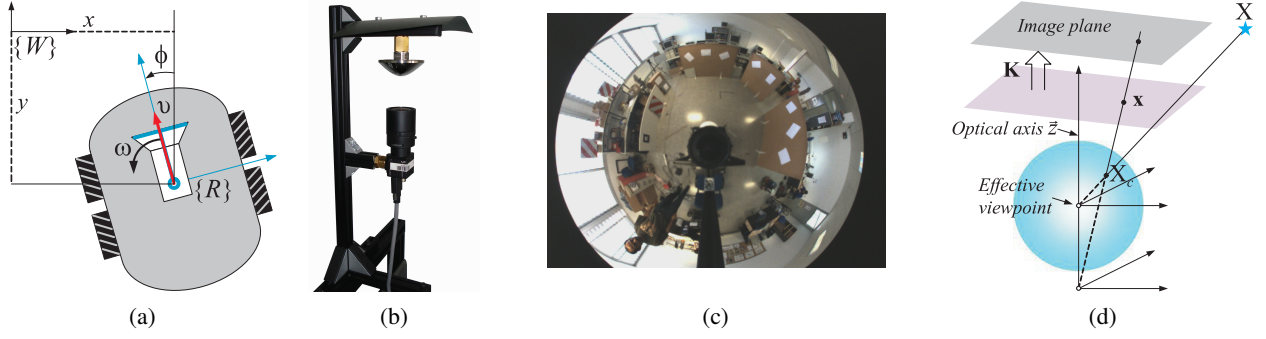


Figure 2: Representation of the robot model and the camera model. (a) Robot frame definition with x and y as the world coordinates. (b) Example of central catadioptric vision system. (c) Example of an image captured by a catadioptric system. (d) Generic camera model of central cameras.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\sin \phi & 0 \\ \cos \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (1)$$

being $v(t)$ and $\omega(t)$ the translational and angular input velocities, respectively. We assume that the origin of the robot reference frame coincides with the reference frame associated with a fixed camera mounted on the robot, in such a way that the optical center coincides with the rotational axis of the robot. We consider that perspective and fisheye cameras are arranged looking forward and omnidirectional systems looking upward. Thus, the model (1) also describes the camera motion.

3.2 The generic camera model

The constrained field of view of conventional cameras can be enhanced using wide field of view vision systems such as fisheye cameras or full view omnidirectional cameras. There are some approaches for vision-based robot navigation that exploit particular properties of omnidirectional images, for instance, in [20], the Fourier components of the images, and in [21], the angular information extracted from panoramic views are used. Figures 2(b-c) show an example of a central catadioptric vision system and an image captured by the system. The well known unified projection model works properly for vision systems having approximately a single center of projection, like conventional cameras, catadioptric systems and some fisheye cameras [19]. 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 through K defined as:

$$K = \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where α_x and α_y are the focal length of the perspective camera in terms of pixel dimensions in the x - and y -directions and (u_0, v_0) are the coordinates of the principal point in pixels. In this work we assume that the camera is calibrated [22], which allows us to use the representation of the points on the unitary sphere. Refer to figure 2(d) and consider a 3D point $X = [X, Y, Z]^T$. Its corresponding point coordinates on the sphere X_c can be computed from point coordinates on the normalized image plane $x = [u, v]^T$ and the mirror parameter ξ as follows:

$$\begin{aligned} X_c &= (\eta^{-1} + \xi) \bar{x}, \\ \bar{x} &= \begin{bmatrix} x^T & \frac{1}{1+\xi\eta} \end{bmatrix}^T, \end{aligned} \quad (2)$$

where $\eta = \frac{-\gamma - \xi(u^2 + v^2)}{\xi^2(u^2 + v^2) - 1}$, $\gamma = \sqrt{1 + (1 - \xi^2)(u^2 + v^2)}$. The parameter η is the norm of the 3D point divided by its depth ($\eta = \|X/Z\|$) and it can be computed from point coordinates and depending on the type of sensor. The parameter ξ encodes the nonlinearities of the image formation in the range $\xi \leq 1$ for catadioptric vision systems and $\xi > 1$ for fisheye cameras.

3.3 Multi-view geometric constraints

A multi-view geometric constraint is a mathematical entity that relates the projective geometry between two or more views. The homography model, the epipolar geometry (EG) and the trifocal tensor (TT) have been used for visual control in the literature for the pose regulation problem, e.g., [23], [24], [15]. As opposed to the path following problem, a desired robot configuration is intended to be reached in these works. Also, the homography model and the EG have been used in the context of visual path following for position-based schemes relaxed to an Euclidean reconstruction from the geometric constraint decomposition [5], [6]. We selected EG and

TT in this study, and not the homography, because they provide more general representations of the geometry between cameras in 3D scenes and they are used in our image-based approach.

Next, the estimations of the epipolar geometry and the trifocal tensor from images of a generic camera are briefly described.

3.3.1 The epipolar geometry

The fundamental epipolar constraint is analogue for conventional cameras and central catadioptric systems if it is formulated in terms of rays emanating from the effective viewpoint. Let \mathbf{X}_c and \mathbf{X}_t be the coordinates of a 3D point \mathbf{X} projected onto the unit spheres associated to a current frame F_c and a target frame F_t . The epipolar constraint is then expressed as follows:

$$\mathbf{X}_c^T \mathbf{E} \mathbf{X}_t = 0, \quad (3)$$

being \mathbf{E} the essential matrix relating the pair of normalized cameras. Normalized means that the effect of the known calibration matrix has been removed and geometric points are obtained. The essential matrix can be computed linearly from a set of corresponding image points projected to the sphere using a classical method [25]. The points lying on the baseline and intersecting the corresponding virtual image plane are called epipoles: current epipole $\mathbf{e}_c = [e_{cx}, e_{cy}, e_{cz}]^T$ and target epipole $\mathbf{e}_t = [e_{tx}, e_{ty}, e_{tz}]^T$. They can be computed from $\mathbf{E} \mathbf{e}_c = \mathbf{0}$ and $\mathbf{E}^T \mathbf{e}_t = \mathbf{0}$.

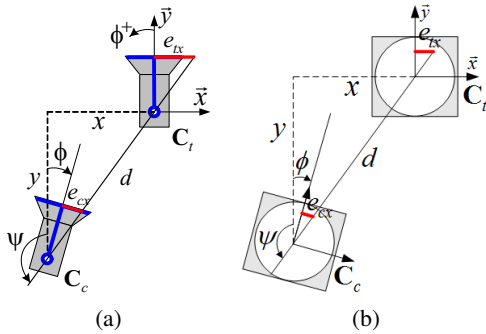


Figure 3: Setup of the epipolar geometry in the x - y plane, eye's-bird view. (a) Perspective cameras. (b) Omnidirectional cameras.

Figure 3 presents the setup of the EG constrained to planar motion, so that, an upper view of two camera configurations is shown. For any case, perspective and omnidirectional cameras, the epipoles give the translation direction between both cameras as depicted in figure 3. The first components of the epipoles, e_{cx} , e_{tx} , provide enough information to characterize the geometry of this planar setup, so that, the values of the

other components are not used. A reference frame centered in the target viewpoint is defined as the origin $\mathbf{C}_t = (0, 0, 0)$. Then, the current camera location with respect to this reference is $\mathbf{C}_c = (x, y, \phi)$. Assuming that the camera and robot reference frames coincide, as shown in figure 2(a), the x -coordinate of the epipoles can be written as a function of the robot state as follows:

$$\begin{aligned} e_{cx} &= \alpha_x \frac{x \cos \phi + y \sin \phi}{y \cos \phi - x \sin \phi}, \\ e_{tx} &= \alpha_x \frac{x}{y}. \end{aligned} \quad (4)$$

The parameter α_x is defined for perspective cameras from the matrix of intrinsic parameters. For the case of normalized cameras, i.e., when points on the sphere are used, $\alpha_x = 1$. Cartesian coordinates x and y can be expressed as a function of the polar coordinates d and ψ :

$$x = -d \sin \psi, \quad y = d \cos \psi, \quad (5)$$

with

$$\psi = -\arctan(e_{tx}/\alpha_x), \quad \phi - \psi = \arctan(e_{cx}/\alpha_x) \quad (6)$$

and $d^2 = x^2 + y^2$. It is worth mentioning that the EG is ill-conditioned for planar scenes and it degenerates with short baseline.

3.3.2 The trifocal tensor

The TT encapsulates the geometric relation between three views, independently of the scene structure [26]. This geometric constraint is more robust and without the mentioned drawbacks of the EG. The TT has 27 elements and it can be expressed using three 3×3 matrices $\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$. Consider three corresponding points \mathbf{p} , \mathbf{p}' and \mathbf{p}'' as shown in figure 4(a) and their projection on the unitary sphere \mathbf{X} , \mathbf{X}' and \mathbf{X}'' expressed as $\mathbf{X} = (X^1, X^2, X^3)^T$. The incidence relation between these points is given by:

$$[\mathbf{X}']_{\times} \left(\sum_i X^i \mathbf{T}_i \right) [\mathbf{X}'']_{\times} = \mathbf{0}_{3 \times 3}, \quad (7)$$

where $[\mathbf{X}]_{\times}$ is the common skew symmetric matrix. This expression provides a set of four linearly independent equations [27]. Thus, seven triplets of point correspondences are needed to compute the 27 elements of the tensor.

Consider a setup where images are taken from three different coplanar locations, i.e., with a camera moving at a fixed distance from the ground plane. In this case, several elements of the tensor are zero and only 12 elements are in general non-null. Figure 4(b) depicts the upper view of three perspective cameras placed on the x - y plane with global reference frame

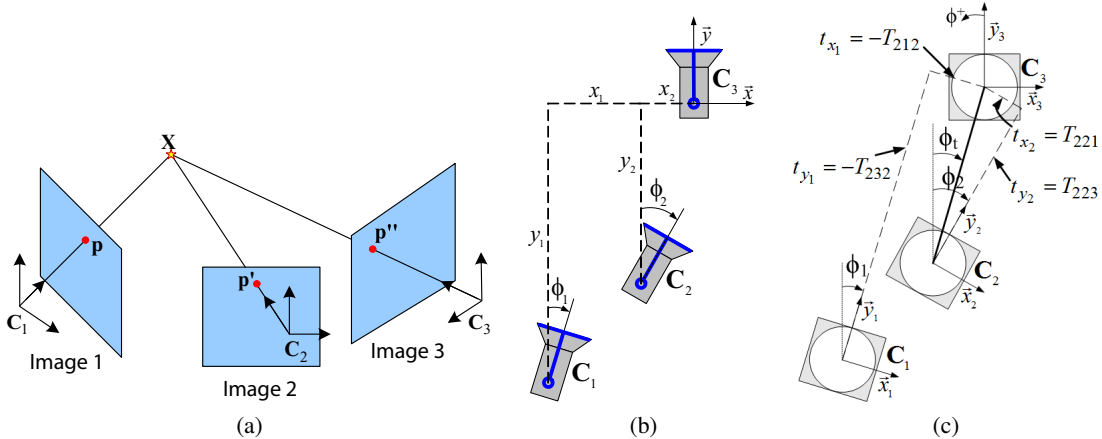


Figure 4: Setup of the trifocal tensor constraint. (a) Point feature correspondences in three images. (b) Eye's-bird view of the geometry in the plane showing absolute locations with respect to a reference frame in C_3 for perspective cameras. (c) Relative locations and their relationships with the tensor elements for omnidirectional cameras, similarly for perspective ones.

in the third view. Hence, the corresponding camera locations are $C_1 = (x_1, y_1, \phi_1)$, $C_2 = (x_2, y_2, \phi_2)$ and $C_3 = (0, 0, 0)$. Figure 4(c) shows the relative locations for the same configurations of cameras but now depicting omnidirectional cameras. The same geometry turns out to be the same TT for both perspective and omnidirectional cameras if the points for the omnidirectional case are projected on the unitary sphere. The TT can be analytically deduced for this setup as done in [15]. For short notation, we use $c\beta = \cos \beta$ and $s\beta = \sin \beta$. Some important non-null elements of the tensor are the following:

$$\begin{aligned} T_{212}^m &= -t_{x_1}, & T_{221}^m &= t_{x_2}, \\ T_{223}^m &= t_{y_2}, & T_{232}^m &= -t_{y_1}, \end{aligned} \quad (8)$$

where $t_{x_i} = -x_i c\phi_i - y_i s\phi_i$, $t_{y_i} = x_i s\phi_i - y_i c\phi_i$ for $i = 1, 2$ and where the superscript m indicates that the tensor elements are computed from metric information. In practice, the estimated tensor has an unknown scale factor and this factor changes as the robot moves. This scaling problem can be avoided by normalizing each element of the tensor as $T_{ijk} = T_{ijk}^e / T_N$, where T_{ijk}^e are the estimated TT elements obtained from point matches and T_N is a suitable normalizing factor. We can see from (8) that T_{212} and T_{232} are non-null assuming that the camera location C_1 is different from C_3 . Therefore, any of these elements is a good candidate as normalizing factor.

4 Navigation strategy

The autonomous navigation stage that allows the robot to follow the learned visual path can be achieved by applying a

non-null translational velocity at the same time that an adequate rotational velocity corrects the lateral deviation from the path. In this section, we describe that both of them, the EG and the TT, provide information about the lateral deviation from the path which can be used for feedback control. In the literature of visual control, these geometric constraints have been used for image-based control of mobile robots for pose regulation (homing) [24], [15], [16], [28], [17]. In these works, both the rotational and the translational velocities are computed from a geometric constraint and they are null when the desired pose is achieved. Although visual path following might be seen as a sequence of homing tasks, such approach would reduce the robot velocity every time an image from the path is approached, which is useless for navigation. Thus, the use of feedback of a geometric constraint has to be adapted to the particular problem. It is worth introducing that the whole path following task can be solved by using exclusively one of the geometric constraints, the EG or the TT, behaving better with the use of this last constraint. Next, two alternative schemes are described and, the benefits of using one or the other scheme are clarified in the section of results.

4.1 Epipole-based navigation

As described previously, the epipoles can be estimated from the essential matrix that encodes the EG between two images (3). Basically, the epipoles are the projection of the optical centers of each camera in the corresponding image. Therefore, the epipoles provide information of the translational direction between optical centers in a two-view configuration. We propose to use only the x -coordinate of the current epipole as feedback information to modify the robot heading, and then, to correct the lateral deviation from the path. As can

be seen in figure 3, e_{cx} is directly related to the required robot rotation to be aligned with the target, assuming that the optical center coincides with the rotational axis of the robot. Consider that the image $I_t(\mathbf{K}, \mathbf{C}_t)$ is one of the key images that belongs to the visual path and $I_c(\mathbf{K}, \mathbf{C}_c)$ is the current image as seen by the onboard camera. A reference frame is attached to the optical center associated to the target image, so that, the x -coordinate of the current epipole is given by (4).

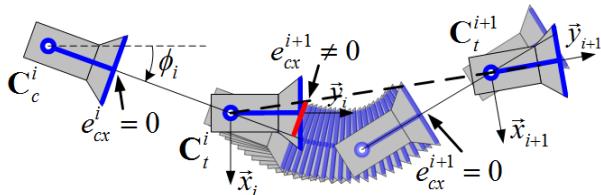


Figure 5: Control strategy based on driving to zero the current epipole.

As can be seen in figure 5, $e_{cx} = 0$ means that the longitudinal axis of the camera-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 time-varying reference. This procedure allows avoiding discontinuous rotational velocity when a new target image is requested 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 as shown later, and to compute a feedforward rotational velocity according to the shape of the path.

As for any visual control scheme, it is a need to express the interaction between the robot velocities and the rate of change of the visual measurements for the control law design, which in this case is given by:

$$\dot{e}_{cx} = -\frac{\alpha_x \sin(\phi - \psi)}{d \cos^2(\phi - \psi)} v + \frac{\alpha_x}{\cos^2(\phi - \psi)} \omega. \quad (9)$$

This equation that expresses the described interaction is derived in the Appendix. The equation will be used in Section 5 to design a controller for ω , while an adequate profile of the translational velocity will be defined.

Other works have exploited the epipolar geometry for image-based visual servoing [24], [16]. The control laws presented in these works are not applicable to the path following problem, given that they are designed specifically to reach a desired pose. Both epipoles e_{cx} and e_{tx} are needed for the controller design. The contribution of the work herein with respect to these previous works is the design of an original feedforward control component. This presents a good piecewise constant behavior of the robot velocities, adequate for

the path following problem.

4.2 Trifocal tensor-based navigation

Consider that we have two images $I_1(\mathbf{K}, \mathbf{C}_1)$ and $I_3(\mathbf{K}, \mathbf{C}_3)$ that are key images of the visual path and the current view of the onboard camera $I_2(\mathbf{K}, \mathbf{C}_2)$. As can be seen in figure 4(c), the element T_{221} of the TT provides direct information of the lateral deviation of the current location \mathbf{C}_2 with respect to the target \mathbf{C}_3 . Let us denote the angle between the \bar{y}_3 axis and the line joining the locations \mathbf{C}_2 and \mathbf{C}_3 as ϕ_t . Given the expression of the element of the tensor $T_{221} = t_{x_2} = -x_2 \cos \phi_2 - y_2 \sin \phi_2$ (8), it can be seen that if $T_{221}^m = 0$, then $\phi_2 = \phi_t = -\tan(x_2/y_2)$.

In such condition, the current camera \mathbf{C}_2 is looking directly toward the target as can be seen in figure 6. In order to benefit from the better properties of the TT in comparison to the EG, we propose a second scheme to compute the rotational velocity by using the element T_{221} as feedback information. The control goal is to drive this element with smooth evolution from its initial value to zero before reaching the next key image of the visual path.

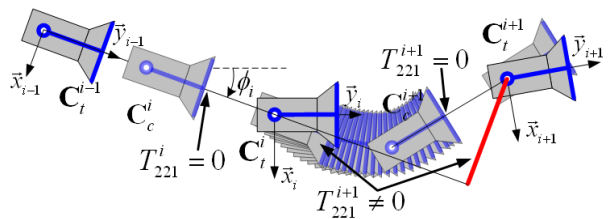


Figure 6: Control strategy based on driving to zero the element of the trifocal tensor T_{221} .

We define a reference tracking control problem in order to avoid discontinuous rotational velocity in the switching of the key images. It is also possible to exploit the a priori information provided by the visual path to compute an adequate translational velocity and also to achieve piecewise constant rotational velocity according to the learned path. In this trifocal tensor-based scheme, the rate of change of the chosen visual measurement only depends on the rotational velocity of the robot as follows:

$$\dot{T}_{221} = T_{223} \omega. \quad (10)$$

Intuitively, the independence of T_{221} on v is because T_{221} is directly proportional to the lateral position error of \mathbf{C}_3 with respect to \mathbf{C}_2 expressed in the reference frame of \mathbf{C}_2 (see figure 4), so that, any longitudinal motion of \mathbf{C}_2 does not produce a change in the lateral error. The derivation of (10) can be seen in the Appendix. This equation will be used in Section 5 to find out the rotational velocity ω . In the path following

problem it is enough only one element of the TT in order to control the deviation with respect to the path. We propose an adequate one-dimensional task function in this case, which is our main contribution with respect to the previous works using the TT [15], [17]. In the former, an over-constrained controller that may suffer from local minima problems is proposed from the TT. This is overcome in the second work by defining a square control system and using measurements from a reduced version of the TT, the so-called 1D-TT. However, none of the control laws of these works is directly applicable for visual path following. An additional contribution of this paper is an appropriate strategy for the particular problem of path following using the TT.

5 Control law for autonomous navigation

The proposed control law is expressed in a generic way in order to drive the robot to follow the visual path using one of the aforementioned navigation strategies. Both alternatives are evaluated in this work and a comprehensive comparison of them is provided in the section of results. Let us define a scalar task function that is defined for each segment of the visual path between two consecutive key images. The task function must be driven to zero for each segment of the visual path. This function represents the tracking error of the current visual measurement m with respect to a desired reference $m^d(t)$:

$$\zeta = m - m^d(t), \quad (11)$$

where the visual measurement m is e_{cx} or T_{221} according to the scheme to be used. This means that the same type of visual measurement is used for the whole task. Although it is not indicated explicitly, the tracking error is defined using the i^{th} key image as target. The following nonlinear differential equation represents the rate of change of the tracking error as given by the robot velocities and it is obtained by taking the time-derivative of the corresponding visual measurement:

$$\dot{\zeta} = f_1(\mathbf{m})v + f_2(\mathbf{m})\omega_t - \dot{m}^d(t), \quad (12)$$

where \mathbf{m} is the whole information that can be obtained from the corresponding geometric constraint (4) or (8). In the case of epipolar feedback $f_1(\mathbf{m}) = -\frac{\alpha_x \sin(\phi-\psi)}{d \cos^2(\phi-\psi)}$ and $f_2(\mathbf{m}) = \frac{\alpha_x}{\cos^2(\phi-\psi)}$ according to (9). In the case of measurements from the TT, $f_1(\mathbf{m}) = 0$ and $f_2(\mathbf{m}) = T_{223}$ as can be seen in (10). We define the desired behavior through the differentiable sinusoidal reference

$$\begin{aligned} m^d(t) &= \frac{m(0)}{2} \left(1 + \cos\left(\frac{\pi}{\tau}t\right) \right), & 0 \leq t \leq \tau, \\ m^d(t) &= 0, & t > \tau, \end{aligned} \quad (13)$$

where $m(0)$ is the value of the visual measurement when a new target image is requested to be reached and τ is a suitable time in which the visual measurement 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-parameter required in the reference can be replaced by the number of iterations of the control cycle. Note that this reference trajectory provides a smooth zeroing of the desired visual measurement from its initial value as can be seen in figure 7.

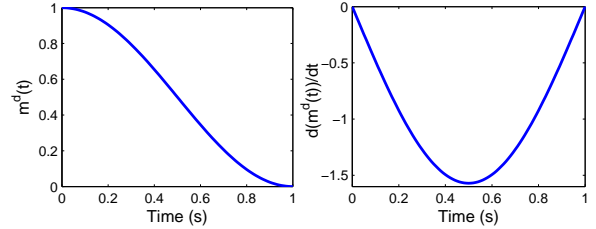


Figure 7: Measurements reference $m^d(t)$ and its time-derivative $\dot{m}^d(t) = -(\pi m(0)/2\tau) \sin(\pi t/\tau)$ for $m(0) = 1$ and $\tau = 1$ s.

The velocity ω_t can be worked out by using input-output linearization of the error dynamics. Thus, the following rotational velocity assigns a new dynamics to the error (12) through the auxiliary input δ_a :

$$\omega_t = -\frac{f_1(\mathbf{m})}{f_2(\mathbf{m})}v + \frac{1}{f_2(\mathbf{m})}(\dot{m}^d + \delta_a). \quad (14)$$

where $\delta_a = -k\zeta$ with $k > 0$ being a control gain. We use the subscript t for the rotational velocity to denote that this velocity has been designed to achieve tracking of the reference $m^d(t)$. The rotational velocity (14) reduces the error dynamics (12) to the closed-loop linear behavior $\dot{\zeta} = -k\zeta$. To verify the well-definition of the rotational velocity, for the EG control we have:

$$\frac{f_1(\mathbf{m})}{f_2(\mathbf{m})} = \frac{-\sin(\phi-\psi)}{d}, \text{ and } \frac{1}{f_2(\mathbf{m})} = \frac{\cos^2(\phi-\psi)}{\alpha_x}.$$

These terms are bounded for any difference of angles $\phi-\psi$, and hence, there are no singularities for the EG control. For the TT-based control a singularity might occur given that $f_2(\mathbf{m})$ is zero if the longitudinal error reaches zero ($T_{223} \propto t_{y_2}$). We prevent the occurrence of such condition by using an adequate key image switching strategy, detailed later in Section 5.2. This strategy is based on aligning the robot toward the next key image in τ seconds, in which an intermediate location d_{\min} is reached. Recall that d_{\min} is assumed as a conservative value of the minimum distance between any pair of consecutive key images. After τ seconds, we make $\omega_t = 0$ and the robot moves with a constant orientation given by $\bar{\omega}$ until an image error starts increasing or the current time exceeds 1.1τ . This strategy prevents the appearance of a sin-

gularity. However, additionally for the TT-based control, we implement that if $f_2(\mathbf{m})$ is below a threshold then $\omega_t = 0$.

Since the control goal of this controller is the reference 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 or feedforward rotational velocity $\bar{\omega}$. We propose to use a priori information extracted from the set of key images to apply an adequate translational velocity and to compute a feedforward rotational velocity. The next section details how this component of the velocity is obtained. So, the complete rotational velocity can be eventually computed as:

$$\omega = k_t \omega_t + \bar{\omega}, \quad (15)$$

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 visual path, however, the complete input velocity (15) behaves more natural around constant values. We will refer as RT to the reference tracking control alone, ω_t (14), and as FF+RT to the complete control, ω (15).

5.1 Exploiting information from the memory

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 information encoded in the visual path 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 [3], [4], [6]. Although the effectiveness of a position-based scheme is not questionable, the effort to achieve a good accuracy may be superior to the obtained gain. Thus, we prefer a qualitative measure of the path curvature. Let us define a visual measurement computed from consecutive pairings/triplets of key images with target in the i^{th} key image:

$$m^{ki} = e_{cx}, \text{ or } m^{ki} = T_{221}/T_{223},$$

where the superscript ki stands for key image. Figure 8 shows that m^{ki} is related to the orientation of the camera in the $(i-1)^{th}$ frame with respect to the i^{th} frame, so that, m^{ki} gives a notion of the curvature for each segment of the path. In order to simplify the notation, we have not used a subscript to denote the i^{th} segment of the path, but recall that the visual measurement m^{ki} is computed for each segment between key images with target in the i^{th} one.

We propose a smooth change of the translational velocity depending on the curvature of the path encoded in m^{ki} . Thus, the translational velocity is computed through the following mapping for every segment between key images:

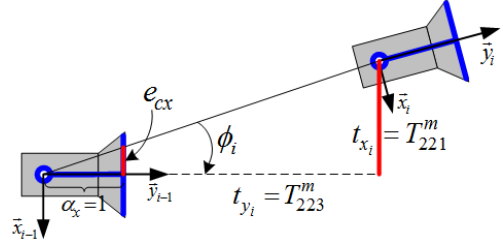


Figure 8: Notion of curvature of the path given by the visual measurement $m^{ki} = e_{cx}$ or $m^{ki} = T_{221}/T_{223}$ for two consecutive key images with target in the i^{th} key image.

$$v = \frac{v_{\max} + v_{\min}}{2} + \frac{v_{\max} - v_{\min}}{2} \tanh\left(\frac{\pi}{2} - \frac{|m^{ki}/d_{\min}|}{\sigma}\right), \quad (16)$$

where σ is a positive parameter that determines the distribution of the velocities between the user-defined limits $[v_{\min}, v_{\max}]$. Figure 9 depicts the mapping for the translational velocity for different parameters d_{\min} and σ . It can be seen that v is high (according to v_{\max}) if m^{ki} is small, which corresponds to a small curvature, and v is low (according to v_{\min}) if m^{ki} is large, i.e., a large curvature is estimated.

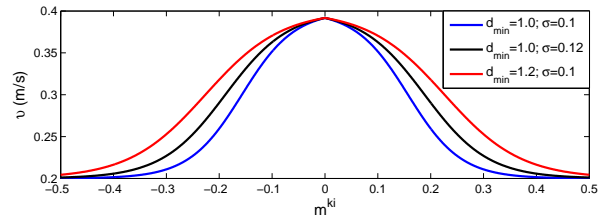


Figure 9: Mapping for the translational velocity as a function of the level of curvature encoded in m^{ki} . The limits of the velocity for the plots are set to $v_{\min} = 0.2$ m/s, $v_{\max} = 0.4$ m/s.

The curvature of the path is related to the ratio ω/v , which is proportional to $m^{ki} = t_{x2}/t_{y2}$ as shown in figure 8. Therefore, once a translational velocity (v) is set from (16) for each key image, its value can be used to compute the nominal velocity $\bar{\omega}$, proportional to the visual measurement m^{ki} , as follows:

$$\bar{\omega} = \frac{k_m v}{d_{\min}} m^{ki}, \quad (17)$$

where $k_m < 0$ is a constant factor to be set. On the one hand, we assume that the commanded translational velocity can be precisely executed by the robot and there is not a feedback component for this velocity. On the other hand, the nominal rotational velocity by itself is able to correct the robot's orientation in order to approximately follow the path. However, the

required closed-loop correction is introduced by the reference tracking control (14) in the complete control (15).

5.2 Timing strategy and key image switching

The proposed control method is based on taking to zero the visual measurement m before reaching the next key image, 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 (v) for each key image as follows:

$$\tau = \frac{d_{\min}}{v}. \quad (18)$$

By running the controller (14) with the reference (13), the time τ and an appropriate control gain k , the robot is oriented toward the target and it reaches an intermediate location determined by d_{\min} . In the best case, when d_{\min} coincides with the real distance between key images, the robot reaches the location associated to the corresponding key image. In order to achieve an adequate correction of the longitudinal error for each key image, if $t > \tau$ the reference (13) is maintained in zero ($m^d(t) = 0$) and $\omega_t = 0$ until the image error starts to increase or the current time exceeds 1.1τ . This means that after τ seconds, we allow the robot moving at a constant rotational velocity given by $\bar{\omega}$ for at most 10% of the time τ . During this time the image error might start to increase and then, a new target image must be given. Otherwise, the condition $t > 1.1\tau$ determines the key image switching. The *image error* is defined as the mean squared error between the r corresponding image points of the current image ($\mathbf{p}_{i,j}$) and points of the next closest target key image (\mathbf{p}_j), i.e.:

$$\epsilon = \frac{1}{r} \sum_{j=1}^r \|\mathbf{p}_j - \mathbf{p}_{i,j}\|, \quad (19)$$

where r is the number of used corresponding points. Similarly to the behavior reported in [7], the image error decreases monotonically until the robot reaches each target image by using the proposed controllers. In our case, the switching condition for the next key image is defined by the increment of the image error, which is verified by using the current and the previous difference of instantaneous values of the image error.

In order to clarify the steps to implement the proposed approach, they are summarized in the Algorithm 1.

The proposed approach outperforms existing approaches of visual navigation in the literature given that it generates the continuity of the rotational velocity when a new target image is given and its applicability is broadened to any camera obeying approximately a central projection. Moreover, the use of a geometric constraint allows the gathering and indirect filtering of many point features into a single measurement. This

Off-line computation

Input parameters: $v_{\min}, v_{\max}, d_{\min}, \sigma, k_m$.

1. Calibrate camera to obtain \mathbf{K} and mirror parameters.
2. Estimate visual measurements m^{k_i} from key images.
3. Compute v for segments between key images from (16).
4. Compute $\bar{\omega}$ for segments between key images from (17).
5. Compute τ for segments between key images from (18).

On-line computation

Input parameters: d_{\min}, k_t (For EG: d, α_x , where $\alpha_x = 1$ for omnidirectional vision).

Output: robot velocities (v, ω).

for initial key image **to** final key image **do**

$t = 0$;

Estimate initial measurement $m(0)$;

while $t \leq 1.1\tau$ **do**

EG: Estimate epipoles (4) and angles (6);

TT: Estimate the TT (8) and normalize it;

Compute $m^d(t)$ (13) and $\dot{m}^d(t)$;

Compute the tracking error ξ (11);

Compute the RT control ω_t (14);

If selected, compute the FF+RT control ω (15);

Smooth the changes of v from the off-line stage;

if $t > \tau$ **then**

$\omega_t = 0$;

if ϵ (19) is increasing **then**

break while;

end if

else

Increase t ;

end if

end while

end for

Algorithm 1: Summary of the proposed visual navigation scheme.

scheme, being an image-based scheme, does not require the extraction of pose parameters, which limits the effect of noise.

6 Experimental evaluation

In this section, we present some simulations results of the proposed navigation schemes and a fair comparison of them is developed. We use the generic camera model [19] to generate synthetic images from the 3D scene of figure 10. The scene consists of the 12 corners of 3 rectangles. In our preliminary work [13], simulation results are analyzed for fisheye cameras. Currently, we focus on comparing the proposed control schemes using a central catadioptric system looking upward.

A set of key images is obtained according to the motion of the robot through the predefined path. Some examples of camera locations associated to key images are shown in figure 10. The learned path starts in the location $(5, -5, 0^\circ)$ and finishes just before closing the 54m long loop. The camera

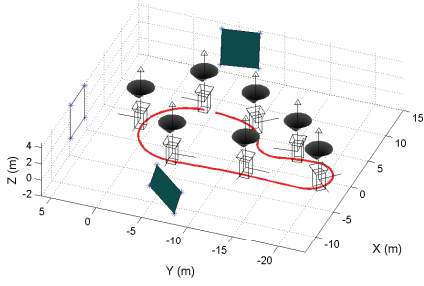


Figure 10: 3D scene and predefined path showing some locations of a central catadioptric camera looking upward.

parameters are used to compute the points on the sphere from the image coordinates as explained in Section 3.2.

6.1 EG-based control vs TT-based control

A comparison of the proposed approach using different visual measurement under the same conditions is carried out through the reference tracking control (RT control of (14)). Hypercatadioptric images of size 800×600 with parameters $\alpha_x = 742$, $\alpha_y = 745$, $u_0 = 400$, $v_0 = 300$, all of them in pixels and $\xi = 0.9662$ are used. A typical 8-point algorithm has been used to estimate the essential matrix and then the epipoles, while a basic 7-point algorithm has been used to estimate the trifocal tensor [26]. The setup of the visual path for this comparison is: 28 key images randomly separated along the predefined path from 1.8m to 2.0m, so that a minimum distance $d_{\min} = 1.75\text{m}$ is assumed. A Gaussian noise with standard deviation of 1 pixel is added to the image coordinates. The translational velocity is bounded between $v_{\min} = 0.2\text{ m/s}$ and $v_{\max} = 0.4\text{ m/s}$ and it is computed according to (16) with $\sigma = 0.1$. It is clear that the velocity for reference tracking (14) reduces the error dynamics to the linear behavior $\dot{\zeta} = -k\zeta$, which exhibits an exponentially stable behavior. Then, the RT control must ensure convergence of the error before the time τ . In order to accomplish such constraint, the control gain must be related to the time τ . Thus, once τ is computed from (18), the control gain k can be set as $k = 12.5/\tau$.

We can see in figure 11 that the resultant path of the autonomous navigation stage is almost similar to the learned one for any of both approaches. Although the initial location is out of the learned path, the robot gets close to the path just in the second key image. We have included labels at the corresponding positions along the path for 60s, 100s and 140s in order to give a reference when observing the next figures.

Figure 12 provides a quantitative comparison of the performance of both approaches through the Cartesian position error and the angular error with respect to the reference path, as

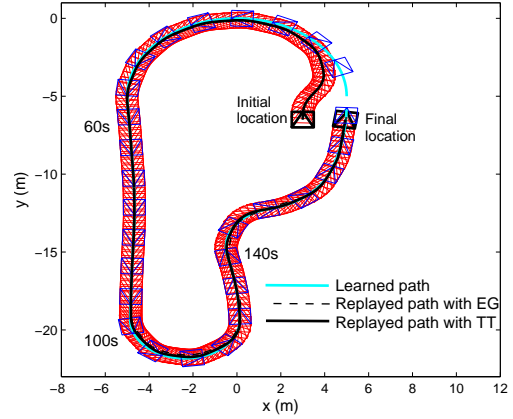


Figure 11: Resultant paths using the RT control and measurement given by the EG and the TT. The distribution of 28 key images is shown over the track traced using the TT control.

well as the image error with respect to the key images. It can be seen that both controls exhibit a comparable performance with a light superiority of the TT control. However, the real superiority of the TT control can be seen in figure 13, where the computed rotational velocity is shown for both TT and EG controls. The velocity given by the TT control is much less affected by the image noise. Notice the occurrence of undesirable large peaks in the rotational velocity given by the EG control in some key image switching, which are due to the short baseline problem of the epipolar constraint. This issue can be also appreciated in the reference tracking performance of figure 14. Although both controls are able to drive the corresponding measurement to track the desired trajectory, the TT control outperforms the EG control given its robustness against image noise.

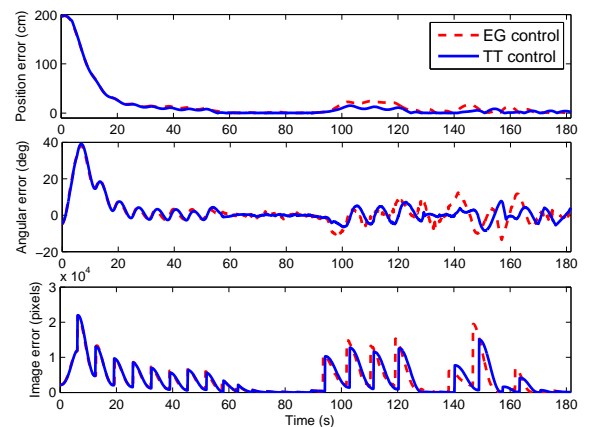


Figure 12: Performance comparison using the RT control and measurement given by the EG and the TT.

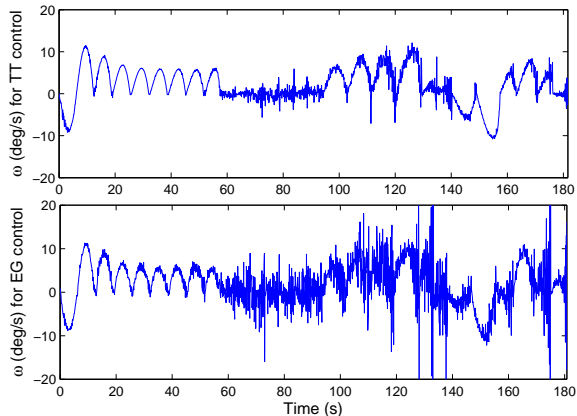


Figure 13: Computed rotational velocities using the RT control and measurement given by the TT and the EG.

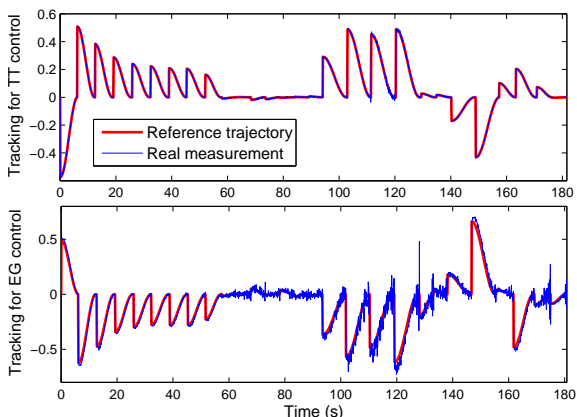


Figure 14: Reference tracking performance, where the reference trajectory is given by (13). The RT control is used and the measurements are given by the TT and the EG, respectively.

6.2 RT control vs FF+RT control

In the sequel of the simulations results, the TT control is used given its superior performance according to the previous results. In this section, a comparison of the RT control alone (14) with respect to the FF+RT control (15) is presented. In order to show the performance of the scheme with different imaging systems, paracatadioptric synthetic images of size 1024×768 pixels are used for the rest of the simulations. The vision system parameters are $\alpha_x = 950$, $\alpha_y = 954$, $u_0 = 512$, $v_0 = 384$, all of them in pixels and $\xi = 1$. In this case, 36 key images are distributed randomly along the learned path. The distance between consecutive key images is between 1.42m and 1.6m, in such a way that a minimum distance $d_{\min} = 1.4$ m is assumed. The image coordinates are

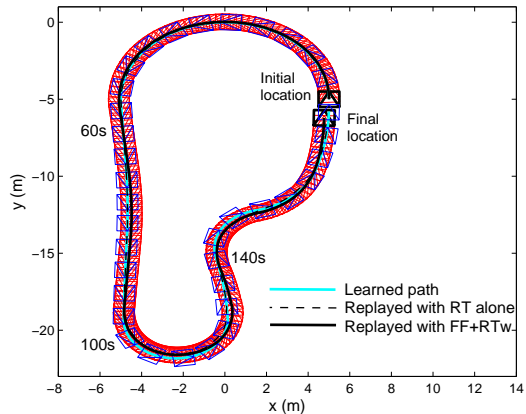


Figure 15: Resultant paths using the controls RT and FF+RTw (weak component of RT) with the TT as measurement. The distribution of 36 key images is shown over the track traced using the FF+RTw control.

affected by an image noise of standard deviation of 1 pixel. The same limits of the translational velocity $v_{\min} = 0.2$ m/s and $v_{\max} = 0.4$ m/s are used in (16) with $\sigma = 0.1$. For the case of the FF+RT control, the weighting factor k_t on the RT control introduced in (15) is set to $k_t = 0.003$, in such a way that its effect is *weak* and the major effect on the control is given by the feedforward term $\bar{\omega}$. We refer to such case FF+RTw.

It can be seen in figure 15 that the path following task for both cases of control RT and FF+RTw is successfully accomplished starting from an initial location on the path. It is worth noting that FF+RTw control, where the major contribution is given by $\bar{\omega}$, is not efficient when the initial robot position is far from the path. However, a suitable k_t allows both controls to be combined accordingly as shown in the next section. Although a small k_t delays convergence toward the path, the feedback component given by $k_t \omega_t$ is enough to achieve convergence from small deviations. Figure 16 presents the plots of the Cartesian position error, the angular error and the image error in order to compare the performance of the RT and the FF+RTw controls. We can see that a comparison from these results is not completely conclusive, given that the maximum position error is obtained using the RT control while the mean position error is larger for the FF+RTw control (10.6cm vs 7cm). Both controls have their maximum errors during the sharpest curves, where the image error is larger for the RT control. However, although the performance of the RT control and the FF+RTw control are actually comparable, the real benefit of the FF+RTw control can be appreciated in the profile of the computed rotational velocities shown in figure 17. The nominal or feedforward value $\bar{\omega}$ allows the robot to achieve a piece-wise constant rotational velocity

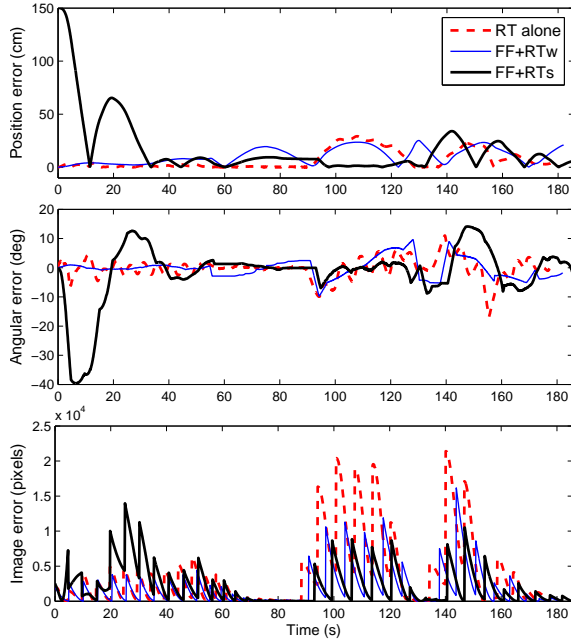


Figure 16: Performance comparison using the RT and FF+RT controls with the TT as measurement. The FF+RT control is evaluated using a weak component of the RT control (FF+RTw) as well as a strong component (FF+RTs). In the case of the FF+RTs control, the robot starts out of the path (see figure 18).

with a mounted small component due to ω_t , which corrects small deviations from the reference path. The same figure 17 presents the varying translational velocity as given by (16) for the whole path. Notice that the velocity is lower from 0 to 55s than from 56s to 95s, which corresponds to the initial curve and the straight segment of the path respectively. This means that the commanded translational velocity corresponds to the level of curvature of the path. We have to point out that the translational velocity computed in the off-line stage from (16) changes in discrete values and these changes must be made smooth to avoid undesired accelerations. We have implemented a sigmoidal transition to smooth the changes of v at key image switching.

6.3 Combining feedforward and reference tracking controls

As said previously, the weighting factor k_t allows the combination of the feedforward and reference tracking controls in such a way that the complete control given by (15) is also efficient when the robot is far from the reference path defined by the key images. The same setup of vision system and visual path of the previous section is used. The weighting factor k_t

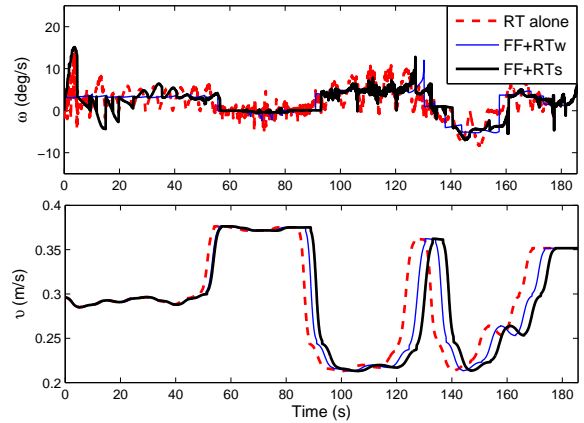


Figure 17: Computed robot velocities given by the RT and FF+RT controls using a weak component of the RT control (FF+RTw) as well as a strong component (FF+RTs), with the TT as measurement.

on the RT control ω_t is set to $k_t = 0.1$, in such a way that the component of the RT control is *strong*. We refer to this case as FF+RTs. Figure 18 shows that the combined approach is able to drive the robot to the reference path from an initial location significantly far from the path. In figure 16, it can be seen that the Cartesian position error and the angular error are around zero after 40s, i.e., after the 6th key image. Around 140s, when the robot traverses a sharp curve, the error increases slightly and the RT component of the control corrects the deviation. The effect of this component can be seen in the computed rotational velocity of figure 17. The velocity keeps the desirable piece-wise constant profile observed in the FF+RTs control of figure 17 when the position error is low, while the RT component modifies the profile when the error is large, as at the beginning of the navigation. It can be seen that the control scheme is independent of the translational velocity. Although the robot starts out of the path, the same profile of translational velocity is used like in the previous case where the robot starts on the path and the commanded task is efficiently accomplished. It is worth noting the possibility of designing an adaptive controller which modifies the parameter k_t as a function of the visual measurement, however, this is left as future work.

In order to show the behavior of the visual information in the image plane, we present the motion of some point features along the navigation in figure 19(a). It is appreciable the advantage of using a central catadioptric system looking upward, which is able to see the same scene during the whole navigation. The snapshots of figure 19(b) show the point features of key images and the current point features at key image switching. The snapshot on the left corresponds to the 7th key image (located in the initial curve) and the right snapshot cor-

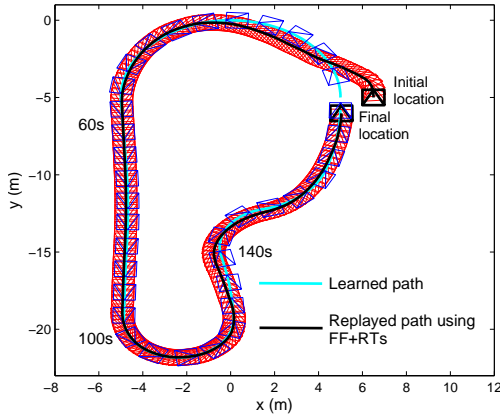


Figure 18: Resultant path using a combination of the feedforward and reference tracking controls, with a strong component of the last (FF+RTs) and the TT as measurement. The distribution of 36 key images is also shown.

responds to the 17th key image (located in the long straight part of the path). As expected, an image error remains when a key image is located along a curve. It is worth noting the effectiveness of the estimation of the TT through points on the sphere obtained from coordinates in the image. Figure 20 presents an example of the projection on the sphere of a triplet of the images used for the navigation.

The proposed navigation scheme in the same form FF+RTs (combination of feedforward control and a strong component of reference tracking) is evaluated against image noise. The TT is used as measurement and the initial location is on the path, in such a way that deviations from the path are effect of the noise. A Gaussian noise with different standard deviation is added to the coordinates of point features. The mean of the absolute values of position and angular errors shown in figure 21 are obtained by averaging over 50 Monte Carlo (MC) runs for each noise level. As expected, there is a tendency to increase the error with the increment of noise level, however, the path following task is always accomplished in such conditions.

6.4 Evaluation using a dynamic simulator

An experimental evaluation has been conducted with the widely used Webots simulator. The set-up consists in a Pioneer 2 robot carrying a single perspective camera, included the dynamics of the robot, not involved in the previous simulations. The environment is made of textured walls on which the feature points used for the evaluation of the trifocal tensor are found. The images taken by the robot camera are 640×480 . A set of 80 key images is previously generated by driving the robot along an arc of circle path. Two key images and the

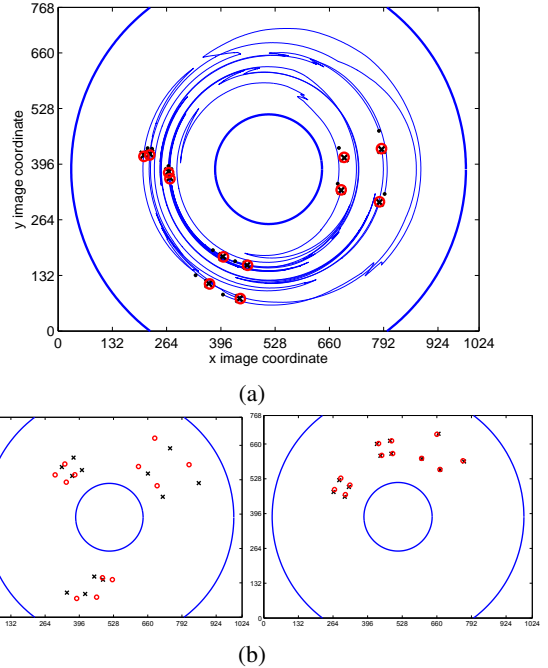


Figure 19: Example of the synthetic visual information: (a) Motion of the points in the images for the navigation of figure 18. The markers are: initial image “.”, final key image “O”, final reached location “×”. (b) Snapshots of current image “×” and the corresponding key image “O”.

current image observed by the robot-camera are used at each iteration for the estimation of the TT. The robot is settled at an initial location with a small lateral deviation from the reference path. We have fixed the translational velocity to 0.05 m/s along the navigation.

In each key image selected along the circular path (the visual memory), a set of Shi-Tomasi points is detected. Then, to produce stable and consistent matchings for their posterior use in TT estimation, the following process is applied: the points detected in key frame $k - 2$ are tracked by the Lucas-Kanade point tracker in frames $k - 1$ and k ; also, the points tracked in frame k are tracked back in frame $k - 1$. This way, if the results of the point tracking from frame $k - 2$ to frame $k - 1$ are quite distant from the results of point tracking from frame $k - 2$ to k and back to $k - 1$, they are simply discarded. This simple algorithm produces sets of matched points for all consecutive groups of three key frames.

For the navigation in this visual memory, we have assumed that the localization problem is solved, i.e. the closest key frame associated to the first robot position is known. Then, we use the algorithm described in Section 5: given the current key frame k and the current image taken by the robot, we track the interest points associated to frame k in the current image, and respectively from key frames $k - 1$ and $k + 1$ (to apply

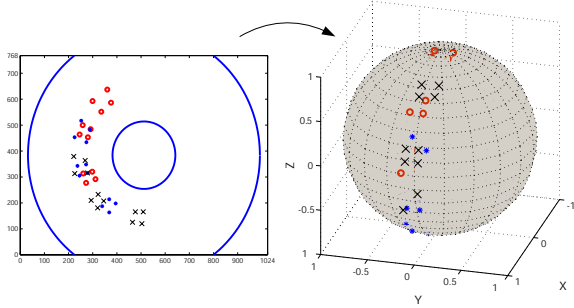


Figure 20: Example of a triplet of images projected on the unitary sphere.

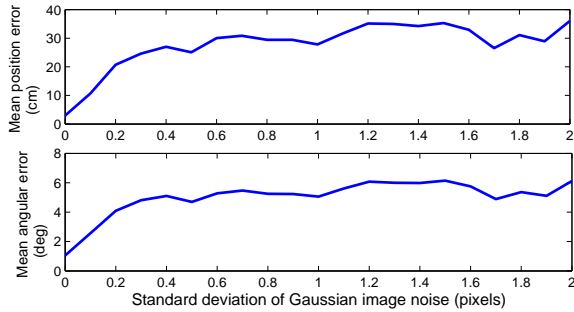


Figure 21: Robustness against image noise of the proposed scheme in the form FF+RTs with initial location on the path.

a consistency test similar to the one described above for the construction of the visual memory); the TT is estimated under the RANSAC algorithm with the matched points. Finally, the computed angular velocity is applied to the robot.

The controller used in the dynamic simulation is in the form of FF+RTs control. Figure 22 shows the performance of our scheme for the visual path following task. Both the learned reference path and the replayed one are shown. The learned path is followed really close during the whole navigation. Figure 23 depicts the rotational velocity given by the controller. It can be seen that the rotational velocity remains around the constant value given by the feedforward term. The velocity exhibits a kind of sinusoidal behavior because the robot position oscillates a little bit around the path. The image error at each iteration is shown in figure 23. The error presents a good decay for each segment between key images.

As an example of the visual information used in the dynamic simulation, figure 24(a) presents one of the first triplets of images used for the estimation of the TT. From left to right, the first key image, the image currently seen by the robot-camera and the second key image are shown. The corresponding image points between the triplet of images are also shown for each one of them. Figure 24(b) illustrates the model of the environment used. Snapshots of the scene with the robot

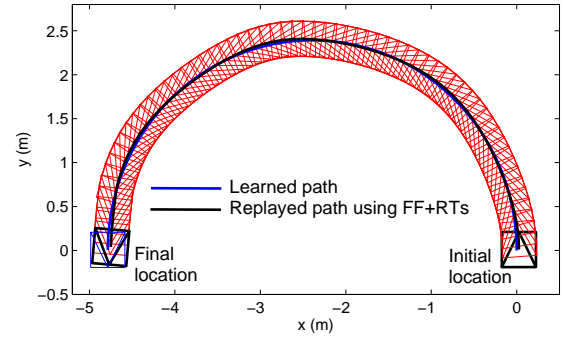


Figure 22: Evaluation of the proposed navigation scheme following an arc of circle path, obtained in the dynamic simulator Webots with the TT as measurement and the FF+RTs control.

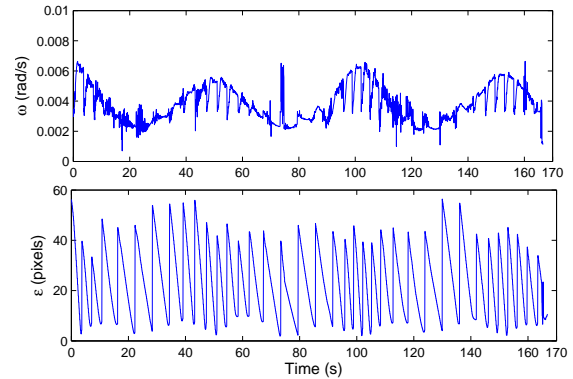


Figure 23: Evaluation of the proposed navigation scheme using a dynamic simulator with the TT as measurement and the FF+RTs control. (a) Computed rotational velocity. (b) Image error.

at three different locations along the navigation are presented. The overall performance of the proposed navigation scheme is proved to be good including dynamic effects. Although dynamic simulation induces for example a small sway motion of the camera, that introduces errors in the model deduced until here from the planar assumption, the presented simulations validate the correct behavior and performance of our proposal.

6.5 Real-world experiments

In order to test the proposed control scheme in a real situation even not for the best option of control, we present an experimental run of a path following task using the EG control. We have used the software platform described in [6] for this purpose. This software selects a set of key images to be reached 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 im-

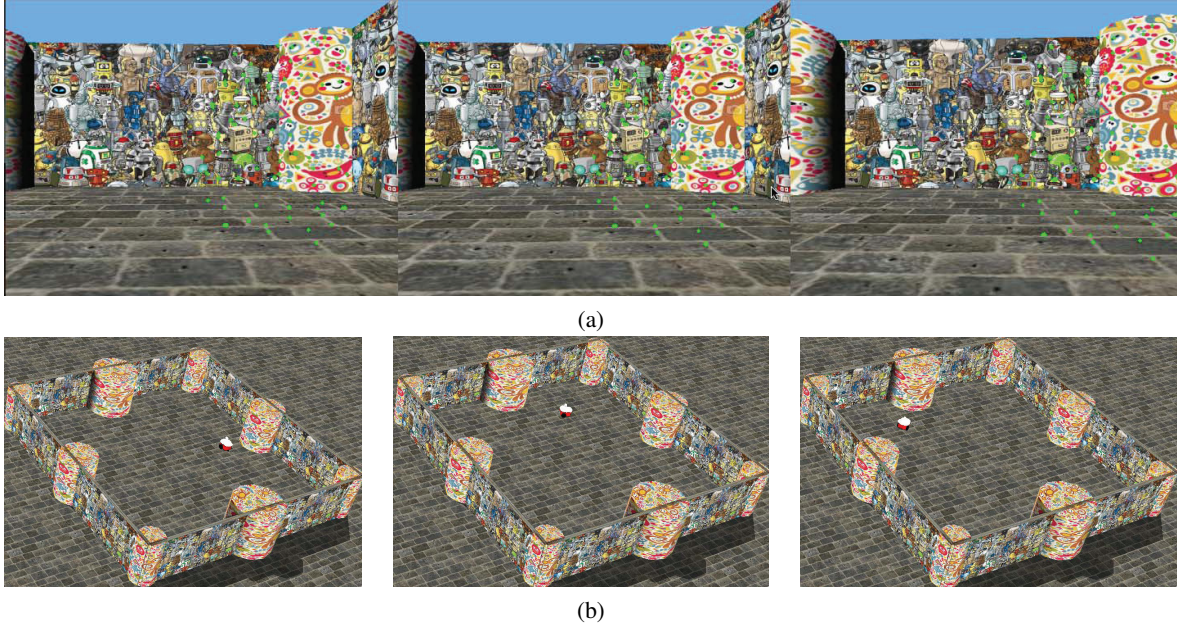


Figure 24: Images from the dynamic simulation of the proposed navigation scheme using Webots. (a) Example of a triplet of images used for the estimation of the TT. From left to right: first key image, current image and second key image. (b) Snapshot of the environment where the robot is shown during the navigation. From left to right: initial location, location around key image 25 and location around key image 45.

ages at each iteration and computes the visual measurement. The interest points are detected in each image with Harris corner detector and then matched by using a Zero Normalized Cross Correlation score. 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. The imaging system consists 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. The camera calibration parameters have been found by using the toolbox [22], however, other options can be used, e.g., [29] and [30]. A constant translational velocity $v = 0.1$ m/s is used along the navigation and a minimum distance between key images $d_{\min} = 0.6$ m is assumed.

Figure 25(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 to ensure a complete correction from an initial robot position that is far from the learned path. The reference path is defined by 12 key images. We can see that after some time, the reference path is reached and followed closely. The robot reaches the reference path by moving forward given that the localization process gives the first key image to be reached in front of the initial location. The computed rotational velocity is presented in the first plot of figure 25(b). The robot follows the visual path until a point

where there are not enough number of matched features (less than 30 matches while the mean was 110). In the same plot, we depict the nominal rotational velocity in order to show that it follows the shape of the path. In the second plot of figure 25(b) 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 figure 25(c) presents a sequence of images as acquired for the robot camera during the navigation.

It is worth commenting that in this experiment the robot has stopped slightly earlier than expected. The reason is the particular situation of the observed scene at the end of the path. In that moment the scene has really little texture, which reduces significantly the amount of matched points. The robot is stopped by the software implementation that we used when the number of points correspondences might jeopardize the good estimation of the epipoles. However, we can state that the proposed navigation scheme is also applicable for long-distance navigation provided that enough number of good matchings are obtained from a robust features matching process.

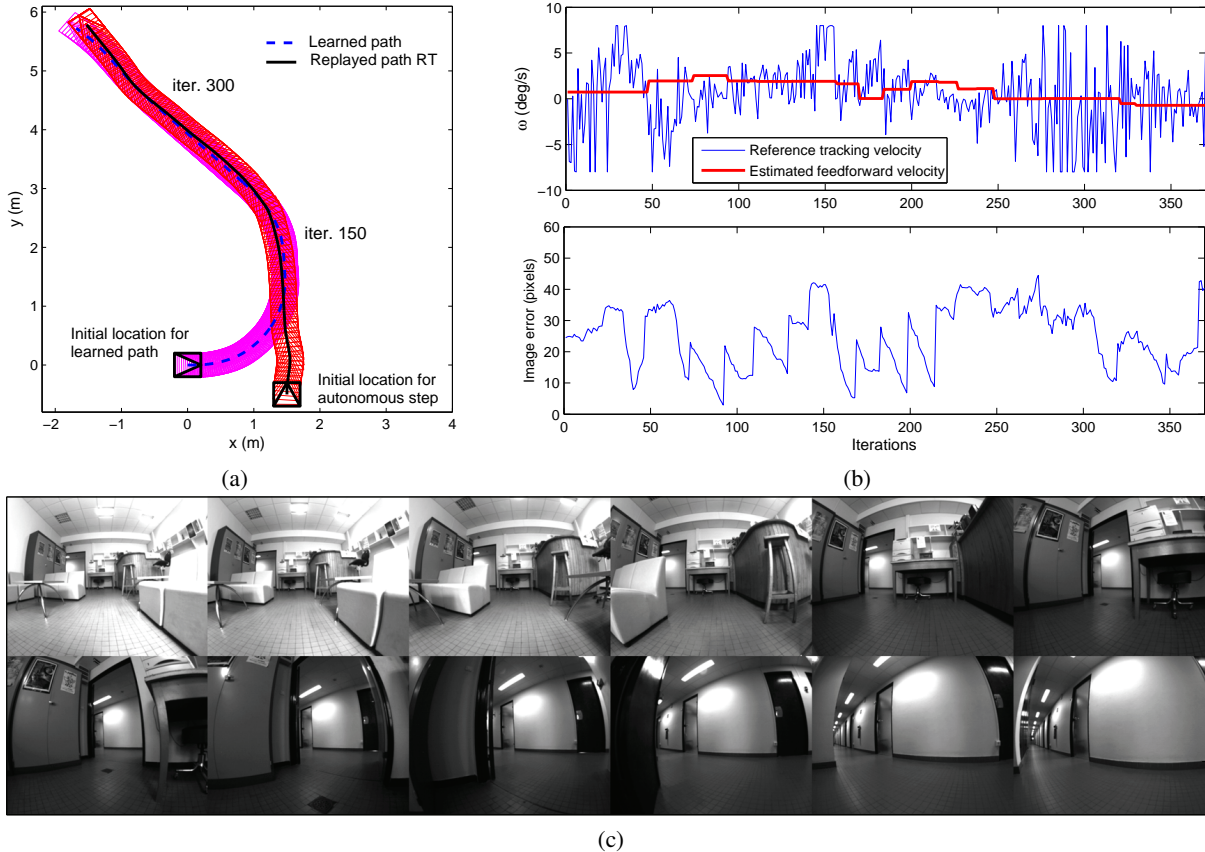


Figure 25: Real-world experiment for indoor navigation using a fish eye camera and the epipoles for feedback. (a) Learned path and resultant replayed path. (b) Computed rotational velocity and image error. (c) Sequence of images during navigation.

7 Conclusions

In this paper, we have developed a control scheme for visual path following, for which, no pose parameters decomposition is carried out. The value of the current epipole or one element of the trifocal tensor is the unique required information by the control law. The use of a geometric constraint allows to gather the information of many point features in only one measurement in order to correct the lateral deviation from the visual path. The approach avoids discontinuous rotational velocity when a new target image must be reached and, eventually, this velocity can be piece-wise constant, which actually represents a contribution with respect to previous works. The translational velocity is adapted according to the shape of the path and the control performance is independent of its value. The proposed scheme relies on the camera calibration parameters to compute the geometric constraints from projected points on the sphere, however, they can be easily obtained using one of the calibration tools currently available in the web. Although both epipolar control and trifocal tensor control solve the navigation problem efficiently, the scheme using feedback

of the trifocal tensor provides better results due to its better conditioning and robustness against image noise. The proposed scheme has exhibited good performance according to simulation results and real-world experiments.

A Appendix. Interaction between visual measurements and robot velocities

The derivation of the expressions (9) and (10) is presented next. These expressions represent the dependence of the rate of change of the visual measurements on the robot velocities. Thus, the time derivative of the x -coordinate of the current epipole (4) after simplification is given by:

$$\dot{e}_{cx} = \alpha_x \frac{\dot{x}y - x\dot{y} + \dot{\phi}(x^2 + y^2)}{(y \cos \phi - x \sin \phi)^2}.$$

Using the kinematic model of the camera-robot (1), we have:

$$\dot{e}_{cx} = \alpha_x \frac{-yv \sin \phi - xv \cos \phi + \omega(x^2 + y^2)}{(y \cos \phi - x \sin \phi)^2},$$

and using the polar coordinates (5) and some algebra:

$$\dot{e}_{cx} = \alpha_x \frac{v(-\sin \phi \cos \psi + \cos \phi \sin \psi)/d + \omega}{(\cos \phi \cos \psi + \sin \phi \sin \psi)^2}.$$

Finally, using trigonometry, it turns out the interaction relationship (9):

$$\dot{e}_{cx} = -\frac{\alpha_x \sin(\phi - \psi)}{d \cos^2(\phi - \psi)} v + \frac{\alpha_x}{\cos^2(\phi - \psi)} \omega.$$

A similar procedure is followed to obtain the time-derivative of T_{221} according to (8) and using the camera-robot model (1):

$$\begin{aligned} \dot{T}_{221}^m &= -\dot{x}_2 c \phi_2 + x_2 \dot{\phi}_2 s \phi_2 - \dot{y}_2 s \phi_2 - y_2 \dot{\phi}_2 c \phi_2, \\ \dot{T}_{221}^m &= v s \phi_2 c \phi_2 + x_2 \omega s \phi_2 - v s \phi_2 c \phi_2 - y_2 \omega c \phi_2, \\ &= (x_2 s \phi_2 - y_2 c \phi_2) \omega. \end{aligned}$$

The expression in parenthesis corresponds to the relative position between C_2 and C_3 , i.e. $t_{y_2} = T_{223}^m$, so that:

$$\dot{T}_{221}^m = T_{223}^m \omega.$$

Finally, by dividing both sides of the equation by the constant element T_{232} the normalized expression (10) is obtained:

$$\dot{T}_{221} = T_{223} \omega.$$

References

- [1] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *IEEE International Conference on Robotics and Automation*, pages 83–88, 1996.
- [2] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue. Visual navigation using omnidirectional view sequence. In *IEEE International Conference on Intelligent Robots and Systems*, pages 317–322, 1999.
- [3] T. Goedeme, M. Nuttin, T. Tuytelaars, and L. V. Gool. Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236, 2007.
- [4] 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.
- [5] J. Courbon, Y. Mezouar, and P. Martinet. Indoor navigation of a non-holonomic mobile robot using a visual memory. *Autonomous Robots*, 25(3):253–266, 2008.
- [6] J. Courbon, Y. Mezouar, and P. Martinet. Autonomous navigation of vehicles from a visual memory using a generic camera model. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):392–402, 2009.
- [7] Z. Chen and S. T. Birchfield. Qualitative vision-based path following. *IEEE Transactions on Robotics*, 25(3):749–754, 2009.
- [8] A. Diosi, S. Segvic, A. Remazeilles, and F. Chaumette. Experimental evaluation of autonomous driving based on visual memory and image-based visual servoing. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):870–883, 2011.
- [9] A. Cherubini and F. Chaumette. Visual navigation with a time-independent varying reference. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5968–5973, 2009.
- [10] A. Cherubini, F. Chaumette, and G. Oriolo. Visual servoing for path reaching with nonholonomic robots. *Robotica*, 29(7):1037–1048, 2011.
- [11] A. De Luca, G. Oriolo, and C. Samson. Feedback control of a nonholonomic car-like robot. In *Robot Motion Planning and Control*. J. P. Laumond (Ed.), Springer-Verlag, New York, USA, 1998.
- [12] A. Cherubini and F. Chaumette. Visual navigation of a mobile robot with laser-based collision avoidance. *International Journal of Robotics Research*, 32(2):189–205, 2013.
- [13] H. M. Becerra, J. Courbon, Y. Mezouar, and C. Sagüés. Wheeled mobile robots navigation from a visual memory using wide field of view cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5693–5699, 2010.
- [14] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette. A mapping and localization framework for scalable appearance-based navigation. *Computer Vision and Image Understanding*, 113(2):172–187, 2009.
- [15] G. López-Nicolás, J.J. Guerrero, and C. Sagüés. Visual control through the trifocal tensor for nonholonomic robots. *Robotics and Autonomous Systems*, 58(2):216–226, 2010.

- [16] H. M. Becerra, G. López-Nicolás, and C. Sagüés. A sliding mode control law for mobile robots based on epipolar visual servoing from three views. *IEEE Transactions on Robotics*, 27(1):175–183, 2011.
- [17] H. M. Becerra, G. López-Nicolás, and C. Sagüés. Omnidirectional visual control of mobile robots based on the 1D trifocal tensor. *Robotics and Autonomous Systems*, 58(6):796–808, 2010.
- [18] A. Cherubini, M. Colafrancesco, G. Oriolo, L. Freda, and F. Chaumette. Comparing appearance-based controllers for nonholonomic navigation from a visual memory. In *Workshop on Safe navigation in open and dynamic environments - Application to autonomous vehicles*, *IEEE Int. Conf. on Robot. and Autom.*, 2009.
- [19] C. Geyer and K. Daniilidis. An unifying theory for central panoramic systems and practical implications. In *European Conference on Computer Vision*, pages 445–461, 2000.
- [20] E. Menegatti, T. Maeda, and H. Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267, 2004.
- [21] A. A. Argyros, K. E. Bekris, S. C. Orphanoudakis, and L. E. Kavradi. Robot homing by exploiting panoramic vision. *Autonomous Robots*, 19(1):7–25, 2005.
- [22] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, pages 3945–3950, 2007.
- [23] Y. Fang, W. E. Dixon, D. M. Dawson, and P. Chawda. Homography-based visual servo regulation of mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 35(5):1041–1050, 2005.
- [24] G. L. Mariottini, G. Oriolo, and D. Prattichizzo. Image-based visual servoing for nonholonomic mobile robots using epipolar geometry. *IEEE Transactions on Robotics*, 23(1):87–100, 2007.
- [25] R. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.
- [26] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
- [27] R. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, 1997.
- [28] G. López-Nicolás and C. Sagüés. Vision-based exponential stabilization of mobile robots. *Autonomous Robots*, 30(3):293–306, 2011.
- [29] D. Scaramuzza and R. Siegwart. A toolbox for easy calibrating omnidirectional cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701, 2006.
- [30] OpenCV library [Online]. Available: <http://sourceforge.net/projects/opencvlibrary/>.