Comparing appearance-based controllers for nonholonomic navigation from a visual memory

Andrea Cherubini, M. Colafrancesco, G. Oriolo, L. Freda, François Chaumette

Comparing appearance-based controllers for nonholonomic navigation from a visual memory

Andrea Cherubini, Manuel Colafrancesco, Giuseppe Oriolo, Luigi Freda and François Chaumette

Abstract— **In recent research, autonomous vehicle navigation has been often done by processing visual information. This approach is useful in urban environments, where tall buildings can disturb satellite receiving and GPS localization, while offering numerous and useful visual features. Our vehicle uses a monocular camera, and the path is represented as a series of reference images. Since the robot is equipped with only one camera, it is difficult to guarantee vehicle pose accuracy during navigation. The main contribution of this article is the evaluation and comparison (both in the image and in the 3D pose state space) of six appearance-based controllers (one pose-based controller, and five image-based) for replaying the reference path. Experimental results, in a simulated environment, as well as on a real robot, are presented. The experiments show that the two image jacobian controllers, that exploit the epipolar geometry to estimate feature depth, outperform the four other controllers, both in the pose and in the image space. We also show that image jacobian controllers, that use uniform feature depths, prove to be effective alternatives, whenever sensor calibration or depth estimation are inaccurate.**

I. INTRODUCTION

In recent research, mobile robot navigation has been often done by processing visual information [1]. This approach can be useful for navigation in urban environments, where tall buildings can disturb satellite receiving and GPS localization, while offering numerous and useful visual features. The most widespread approaches to visual navigation are the *model-based*, and the *appearance-based* approaches, which we shall briefly recall. Model-based approaches rely on the knowledge of a 3D model of the navigation space. The model utilizes perceived features (e.g., lines, planes, or points), and a learning step can be used for estimating it. Conversely, the appearance-based approach does not require a 3D model of the environment, and works directly in the sensor space. The environment is described by a topological graph, where each node corresponds to the description of a position, and a link between two nodes defines the possibility for the robot to move autonomously between the two positions.

In this work, we focus on appearance-based navigation, with a single vision sensor. The environment descriptors correspond to images stored in an *image database*. A similarity score between the view acquired by the camera and the database images, is used as input for the controller that leads the robot to its final destination (which corresponds to a *goal image* in the database). Various strategies can be used to control the robot during navigation. An effective method is visual servoing [2], which was originally developed for manipulator arms, but has also been used for controlling nonholonomic robots (see, for instance, [3]).

The main contribution of this paper is the comparison between six controllers for nonholonomic appearance-based navigation using monocular vision. In particular we investigate the performance of this controllers both in the image, and in the 3D pose state spaces. The paper is organized as follows. In Sect. II, a survey of related works is carried out. In Sect. III, the problem of appearance-based nonholonomic navigation from a visual memory is defined. Although the scope of this paper is the discussion of the control strategies, in Sect. IV, we outline the image processing and the 3D reconstruction algorithms used in our navigation framework. In Sect. V, we present and illustrate the six controllers. The simulated and experimental results are presented in Sect. VI.

A. Cherubini, M. Colafrancesco, G. Oriolo and L. Freda are with the Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Ariosto 25, 00185 Roma, Italy {cherubini, oriolo, freda}@dis.uniroma1.it

F. Chaumette is with INRIA-IRISA, Campus de Beaulieu 35042, Rennes, France Francois.Chaumette@irisa.fr

II. RELATED WORK

Recent works in the field of appearance-based autonomous vehicle navigation are surveyed hereby. Most of these works [3 – 13] present a framework with these characteristics:

- a wheeled robot with an on-board camera is considered;
- during a preliminary phase, the *teaching phase*, the robot motion is controlled by a human operator, and a set of images is acquired and stored in a database;
- an *image path* to track is then described by an ordered set of *reference images*, extracted from the database;
- during the *replaying phase*, the robot (starting 'near' the teaching phase initial position) is required to repeat the same path;
- the replaying phase relies on a matching procedure (usually based on correlation) that compares the currently observed image with the reference images;
- although the control strategy enabling the robot to track the learned path varies from one work to the other, it relies, in all cases, on the comparison between the current and reference images.

The methods presented hereby can be subdivided in two main areas. In some works, a three dimensional reconstruction of the workspace is used. The other navigation frameworks, instead, rely uniquely on image information.

We firstly survey the works where 3D reconstruction is utilized. In 1996, Ohno and others [4] propose to use the image database to reconstruct the robot pose in the workspace (i.e., position and orientation) which is utilized for control. In [5], a three dimensional representation of the

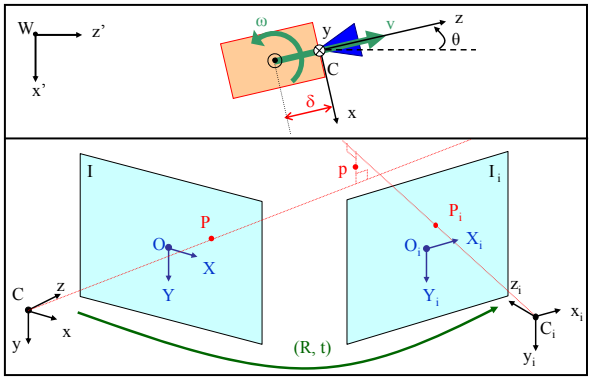

Fig. 1. Relevant variables utilized in this work. Top: mobile robot (orange), equipped with fixed pinhole camera (blue), and applied control variables (v, ω). Bottom: two different views (distinct camera placements) of the same 3-D point p, i.e., in the current (left) and reference (right) images.

taught path is built from the image sequence, and a classic path following controller is used for navigation. Similarly, in [6], pairs of neighboring reference images are associated to a straight line in the 3D workspace, that the robot must track. The epipolar geometry and a planar floor constraint are used to compute the robot heading used for control in [7]. Similarly, in [8], 3D reconstruction is used to improve an omnidirectional vision-based navigation framework.

In general, 3D reconstruction is unnecessary, since moving from one reference image to the next, can also be done by relying uniquely on visual information, as shown in many papers. For instance, in [3], the vehicle velocity commands and the camera pan angle are determined using an image-based visual servoing scheme. In [9], a particular motion (e.g., 'go forward', 'turn left') is associated to each image, in order to move from the current to the next image in the database. In [10], a proportional control on the position of the feature centroid in current and reference images drives the robot steering angle, while the translational velocity is set to a constant value. The controller presented in [11] exploits angular information regarding the features matched in panoramic images. Energy normalized cross correlation is used to control the robot heading in [12]. In [13], a specific image jacobian, relating the change of some image features with the changes in motion in the plane, is used for control.

In summary, a large variety of control schemes has been applied for achieving nonholonomic navigation from a visual memory. However, a comparison between the various approaches has never been carried out. Moreover, in most of the cited articles, the focus has been the qualitative evaluation of the proposed navigation framework in real, complex, environments, without a quantitative assessment of the controller performance. In this paper, we shall compare the performance of six approaches to nonholonomic navigation from a visual memory. The controllers will be assessed using various metrics, both in simulations, and in real experiments. In particular, we will compare the controller accuracy both in the image and in the pose state space, since both are fundamental for precise unmanned navigation.

III. PROBLEM DEFINITION

A. System characteristics

In this work, we focus on a nonholonomic mobile robot of unicycle type, equipped with a fixed pinhole camera. The workspace where the robot moves is planar: $\mathcal{W} = \mathbb{R}^2$. With reference to Fig. 1, let us define the reference frames: world frame $\mathcal{F}_\mathcal{W}\,(W, x', z')$, and image frame $\mathcal{F}_\mathcal{I}(O, X, Y)$ (point O is the image plane center). The *robot configuration* is: $q = [x'\ z'\ \theta]^T$, where $[x'\ z']^T$ is the *Cartesian position* of the robot center in $\mathcal{F}_\mathcal{W}$, and $\theta \in\ ]-\pi, +\pi]$ is the *robot heading* (positive counterclockwise) with respect to the world frame z' axis. We choose $u = [v\omega]^T$ as the pair of control variables for our system; these represent respectively the linear and angular velocities (positive counterclockwise) of the robot. The state equation of the robot is:

$$\dot{q} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} u$$

We also define the camera frame $\mathcal{F}_\mathcal{C}(C, x, y, z)$, shown in Fig. 1 ($C$ is the optical center). The distance between the y axis and the robot rotation axis is denoted by δ. A pinhole camera model is considered; radial distortion is neglected. Hence, the camera intrinsic parameters are the principal point coordinates and the focal lengths in horizontal and vertical pixel size: f_X, and f_Y. In the following, we consider that the camera parameters have been determined through a preliminary calibration phase, although we shall partially relax this assumption later in the paper. Image processing is based on the grey-level intensity of the image, called $I(P)$ for pixel $P = (X, Y)$.

As outlined in Sect. II, our navigation framework relies on a teaching and on a replaying phases. These phases will be described in the rest of this section.

B. Teaching phase

During the teaching phase, an operator guides the robot stepwise along a continuous path. Each of the N *teaching steps* starts at time t_{i-1} and ends at $t_i > t_{i-1}$ $(i = 1, \ldots, N)$. At each step i, the control input u is assigned arbitrarily by the operator. In this work, we assume that throughout teaching, the robot moves forward, i.e., $v > 0$. At the end of each teaching step, the robot acquires a reference image, that we call I_i, and stores it in a database. Visual features are detected in each I_i. We call $\mathcal{F}_{\mathcal{C}i}(C_i, x_i, y_i, z_i)$ and $\mathcal{F}_{\mathcal{I}i}(O_i, X_i, Y_i)$ (see Fig. 1) the N camera and corresponding N image frames associated to the reference configurations q_i reached at the end of each teaching step.

C. Replaying phase

At the beginning of the replaying phase, the robot is placed at the starting position of the teaching phase. During the replaying phase, the robot must autonomously track the path executed during the teaching phase. The task of replaying the taught path is divided into N subtasks, each consisting of zeroing the visual error between the currently acquired image (called I) and the next reference image $(I_1, I_2, \ldots, I_N)$ in

the database. In practice, as soon as the visual error between I and goal image I_i is 'small enough', the subtask becomes that of reaching image I_{i+1}. Both the visual error and the switching condition will be detailed in Sect. V. Throughout replaying, the linear velocity is fixed to a constant value $\bar{v} > 0$, while the angular velocity ω is derived with a feedback law dependent on the visual features. In all six feedback controllers that we have tested, at each iteration of subtask i, ω is based on the feature points matched between the current image I and the reference image I_i.

IV. VISION ISSUES

A. Image processing

During both teaching and replaying, the images acquired by the robot camera must be processed in order to detect feature points. Besides, during the replaying phase, correspondences between feature points in images I and I_i are required to generate the set of matched points which is used to control the robot. In both teaching and replaying phases, we detect feature points with the well known Harris corner detector [14]. Every iteration of the replaying phase relies on image matching between Harris corners in the current image I and in the nearest next reference image in the database I_i. For each feature point P in image I, we use a correlation technique to select the most similar corresponding point P_i in image I_i. For each pair of images (I, I_i), the algorithm returns the n pairs of *matched points* $(P, P_i)_j$, $j = 1, \ldots, n$.

B. Deriving 3D information

In one the control schemes used in this work (i.e., the *robot heading controller*), it is necessary to estimate the camera pose variation (rotation $\mathbf{R}$ and translation $\mathbf{t}$, see Fig. 1) between the current view I and the next reference view I_i during replay. Moreover, in two of the five image jacobian controllers used, the z coordinates in $\mathcal{F}_\mathcal{C}$ (i.e., the *depths*) of the retroperspective projection p of feature points must be estimated. The depths can also be derived from the camera pose variation. The problem of estimating the camera pose variation ($\mathbf{R}$, $\mathbf{t}$) is a typical *structure from motion* problem.

In some works (see, for instance, [5]), the camera pose is estimated by using bundle adjustment methods, which result in long computation processing, unsuitable for on-line use. Here, we have decided to perform on-line 3D reconstruction, by using only the pair of images (I, I_i), instead of I with the whole database. This choice inevitably implies lower computational time to the detriment of the 3D reconstruction accuracy. The technique that we used for camera pose estimation is epipolar geometry (see [15], for further details). Using an estimate of the distance from q to q_i for $\|\mathbf{t}\|$, four alternative solutions $(\mathbf{R}, \mathbf{t})$ can be derived. For each of the four possible pose variations, we use the technique described in [16] to derive the feature point 3D position p, as the midpoint on the perpendicular to the projecting rays in the two camera frames (see Fig. 1). Finally, we select the pose variation $(\mathbf{R}, \mathbf{t})$ with the greatest number of positive depths in both camera frames $\mathcal{F}_\mathcal{C}$ and $\mathcal{F}_{\mathcal{C}i}$, since feature points must lie in front of both image planes.

V. CONTROL SCHEMES

In this section, we describe the characteristics of the six controllers on ω that we have tested in the replaying phase (v is fixed to constant value $\bar{v}$, see Sect. III). In all cases, we consider that subtask i (i.e., reaching image I_i) is achieved, and we consequently switch to reaching image I_{i+1}, as soon as the average feature error:

$$\epsilon_i = \frac{\sum_{j=1}^{n} \|P_j - P_{i,j}\|}{n}$$

is below a threshold τ_ϵ, and starts to rise.

The first feedback law that we will describe, is *pose-based*: the feedback law is expressed in the robot workspace, by using the 3D data derived from image matching as described in Sect. IV-B. The 5 other feedback laws, instead, are *image-based*: both the control task, and the control law are expressed in the image space, by using the well known image jacobian paradigm. In practice, an error signal measured directly in the image is mapped to actuator commands. Two of the 5 image jacobian controllers require camera pose estimation to derive the depth of feature points. For the 3 others, some approximations on the feature depths are used, as will be shown below.

We hereby recall the image jacobian paradigm which is used by the five image-based controllers. The image jacobian is a well known tool in image-based visual servo control [2], which is used to drive a vector of k visual features s to a desired value s^*. It has been previously applied for solving the problem of nonholonomic appearance-based navigation from a visual memory (see, e.g., [3] and [13]). Let us define:

$$u_c = [v_{c,x}\ v_{c,y}\ v_{c,z}\ \omega_{c,x}\ \omega_{c,y}\ \omega_{c,z}]^T$$

the camera velocity expressed in $\mathcal{F}_\mathcal{C}$. The matrix $\mathbf{L}_\mathtt{S}$ relates the velocity of feature s to u_c:

$$\dot{s} = \mathbf{L}_\mathtt{S} u_c \tag{1}$$

For the robot model that we are considering, the camera velocity u_c can be expressed in function of $u = [v\ \omega]^T$ by using the homogeneous transformation:

$$u_c = {}^\mathtt{C}\mathbf{T}_\mathtt{R} u \tag{2}$$

with:

$${}^\mathtt{C}\mathbf{T}_\mathtt{R} = \begin{bmatrix} 0 & -\delta \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}$$

In the following, we will call T_v and T_ω the first and second columns of ${}^\mathtt{C}\mathbf{T}_\mathtt{R}$. Injecting (2) in (1), we obtain:

$$\dot{s} = \mathbf{L}_{\mathtt{S},v} v + \mathbf{L}_{\mathtt{S},\omega} \omega$$

where $\mathbf{L}_{\mathrm{S},v} = \mathbf{L}_{\mathrm{S}} T_v$, and $\mathbf{L}_{\mathrm{S},\omega} = \mathbf{L}_{\mathrm{S}} T_\omega$ are $k \times 1$ column vectors. In order to drive s to the desired value s^*, we set $v = \bar{v}$ and we select as control law on ω:

$$\omega = -{\mathbf{L}_{\mathrm{S},\omega}}^{+} \left(\lambda e + \mathbf{L}_{\mathrm{S},v} \bar{v}\right) \tag{3}$$

where λ is a given positive gain, e is the error $s - s^*$, and ${\mathbf{L}_{\mathrm{S},\omega}}^{+} \in I\!R^{1\times k}$ is the Moore-Penrose matrix pseudoinverse of $\mathbf{L}_{\mathrm{S},\omega}$, i.e., ${\mathbf{L}_{\mathrm{S},\omega}}^{+} = \left({\mathbf{L}_{\mathrm{S},\omega}}^{T} \mathbf{L}_{\mathrm{S},\omega}\right)^{-1} {\mathbf{L}_{\mathrm{S},\omega}}^{T}$.

A. Robot heading controller

The first controller that we tested in this work, the *robot heading controller* (called RH), is based on the 3D information, derived as described in Sect. IV-B. Since the y-axis is parallel to the robot rotation axis, from the matrix $\mathbf{R}$ defining the rotation between the two camera frames, it is trivial to derive the relative heading variation between the two robot configurations $\Delta\theta = \theta - \theta_i$. Then, we apply the control law:

$$\omega = -\lambda \Delta\theta$$

with λ a given positive gain. A similar controller has been used in [7]. In contrast with that work, however, we do not use the planar constraint to derive the 3D pose variation, and we use $\mathbf{R}$, instead of $\mathbf{t}$ (which is usually more affected by noise), to derive the heading value.

B. Image jacobian points controller

In the *image jacobian points controller* (IJP), the visual features used for achieving subtask i are the current image I coordinates of the n matched points:

$$s = [X_1, Y_1, \ldots, Y_n]^T \in I\!R^{2n}$$

Each subtask i will consists of zeroing error:

$$e = [X_1 - X_{i,1}, Y_1 - Y_{i,1}, \ldots, Y_n - Y_{i,n}]^T \in I\!R^{2n}$$

For a normalized perspective camera, the expression of $\mathbf{L}_{\mathrm{P}}$ for a single image point $P\,(X, Y)$ seen in I is:

$$\mathbf{L}_{\mathrm{P}} = \begin{bmatrix} -\frac{1}{z} & 0 & \frac{X}{z} & XY & -1-X^2 & Y \\ 0 & -\frac{1}{z} & \frac{Y}{z} & 1+Y^2 & -XY & -X \end{bmatrix}$$

where z is derived with the method described in Sect. IV-B. By applying the transformation ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{R}}$, we obtain:

$$\mathbf{L}_{\mathrm{P},v} = \begin{bmatrix} \frac{X}{z} \\ \frac{Y}{z} \end{bmatrix} \qquad \mathbf{L}_{\mathrm{P},\omega} = \begin{bmatrix} \frac{\delta}{z} + 1 + X^2 \\ XY \end{bmatrix}$$

If we consider all n matched points between I and I_i, by merely stacking n times vectors $\mathbf{L}_{\mathrm{P},v}$ and $\mathbf{L}_{\mathrm{P},\omega}$, we obtain the two $2n \times 1$ column vectors $\mathbf{L}_{\mathrm{S},v}$ and $\mathbf{L}_{\mathrm{S},\omega}$ to be used in (3)[1].

[1] ${\mathbf{L}_{\mathrm{S},\omega}}^{+}$ is always defined, since:

$${\mathbf{L}_{\mathrm{S},\omega}}^{T} \mathbf{L}_{\mathrm{S},\omega} = \sum_{j=1}^{n} \left[\left(\frac{\delta}{z_j} + 1 + X_j^2 \right)^2 + (X_j Y_j)^2 \right] > 0$$

because $\frac{\delta}{z} + 1 + X^2 > 0$ for all P.

C. Image jacobian points controller with uniform depths

The *image jacobian points controller with uniform depths* (IJPU) is based on an approximation of the model used by the IJP controller. The only difference between the IJPU controller and the IJP controller, is that the depths of all points P_j (which are required for calculating $\mathbf{L}_{\mathrm{S},v}$ and $\mathbf{L}_{\mathrm{S},\omega}$) are assumed identical and set to a fixed value:

$$z_j = \bar{z} \qquad\qquad \forall j = 1, \ldots, n$$

Although this approximation requires $\bar{z}$ to be tuned by the user, depending on the workspace characteristics, and although it can lead to imprecision in the case of sparse 3D points, setting $z_j = \bar{z}$ avoids the need for 3D reconstruction, and consequently spares computational resources. In practice, the IJPU uses an approximation of the interaction matrix similar to the ones commonly used in the visual servoing literature, when pose estimation should be avoided. In fact, it has been shown in many works that a coarse approximation of the image jacobian, without depth estimation, is often sufficient to achieve visual servoing tasks [2], and uniform depths have been successfully used in [17].

D. Image jacobian centroid controller

In the *Image jacobian centroid controller* (IJC), the visual features used for achieving subtask i are the current image I coordinates of the centroid of the n matched points:

$$s = [X_G, Y_G]^T = \frac{1}{n} \sum_{j=1}^{n} [X_j, Y_j]^T \in I\!R^2$$

Each subtask i will consists of zeroing error:

$$e = [X_G - X_{i,G}, Y_G - Y_{i,G}]^T \in I\!R^2$$

For a normalized perspective camera, the expression of $\mathbf{L}_{\mathrm{S}}$ related to the centroid of a discrete set of n image points has been derived in [18] by using image moments:

$$\mathbf{L}_{\mathrm{S}} = \frac{1}{n} \sum_{j=1}^{n} \begin{bmatrix} -\frac{1}{z_j} & 0 & \frac{X_j}{z_j} & X_j Y_j & -1+X_j^2 & Y_j \\ 0 & -\frac{1}{z_j} & \frac{Y_j}{z_j} & 1+Y_j^2 & -X_j Y_j & -\sum_{j=1}^{n} X_j \end{bmatrix}$$

where the z_js are derived with the method described in Sect. IV-B. By applying the transformation ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{R}}$, we obtain:

$$\mathbf{L}_{\mathrm{S},v} = \frac{1}{n} \sum_{j=1}^{n} \begin{bmatrix} \frac{X_j}{z_j} \\ \frac{Y_j}{z_j} \end{bmatrix} \qquad \mathbf{L}_{\mathrm{S},\omega} = \frac{1}{n} \sum_{j=1}^{n} \begin{bmatrix} \frac{\delta}{z_j} + 1 + X_j^2 \\ X_j Y_j \end{bmatrix} \tag{4}$$

These two vectors are used in the control law (3)[2].

[2] ${\mathbf{L}_{\mathrm{S},\omega}}^{+}$ is always defined, since:

$${\mathbf{L}_{\mathrm{S},\omega}}^{T} \mathbf{L}_{\mathrm{S},\omega} = \frac{1}{n^2} \left[\left(\sum_{j=1}^{n} \frac{\delta}{z_j} + 1 + X_j^2 \right)^2 + \left(\sum_{j=1}^{n} X_j Y_j \right)^2 \right] > 0$$

because $\sum_{j=1}^{n} \frac{\delta}{z_j} + 1 + X_j^2 > 0$.

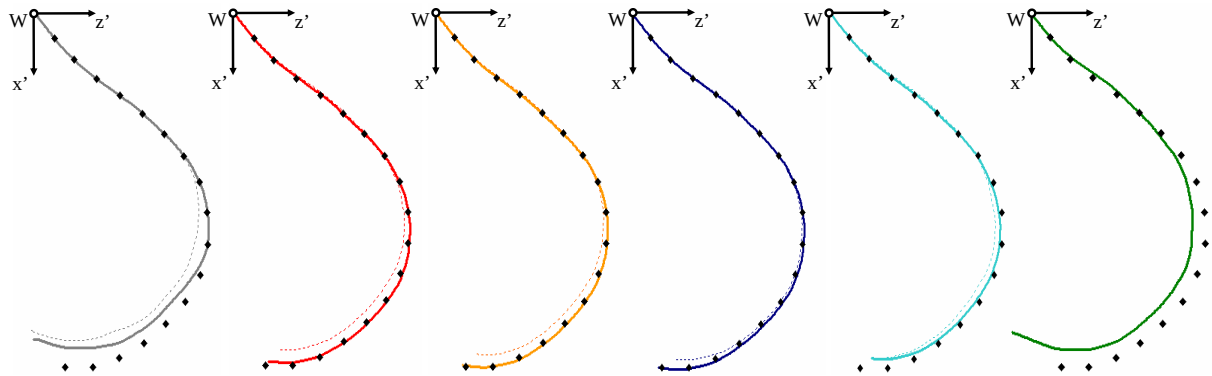

Fig. 2. Robot path in $\mathcal{F}_{\mathcal{W}}$ during Webots navigation, with $N = 17$ reference configurations q_i (black dots), using controllers: RH, IJP, IJPU, IJC, IJCU, AIJCA (left to right), in the experiments with correct (full curves) and coarse (dashed curves) camera calibration.

E. Image jacobian centroid controller with uniform depths

The *image jacobian centroid controller with uniform depths* (IJCU) is based on an approximation of the model used by the IJC controller. The approximation is identical to the one used in IJPU to avoid 3D reconstruction:

$$z_j = \bar{z} \qquad \forall j = 1, \ldots, n$$

with $\bar{z}$ to be tuned according to the workspace characteristics.

F. Approximated image jacobian centroid abscissa controller

From a control viewpoint, since in (3) we control only one degree of freedom (ω), one feature is sufficient for control. In the *approximated image jacobian centroid abscissa controller* (AIJCA), the only visual feature that we use is the abscissa of the centroid of the n matched points in I:

$$s = X_G \in I\!R$$

This choice is reasonable, since the camera optical axis is orthogonal to the robot rotation axis. Each subtask i will then consist of zeroing error:

$$e = X_G - X_{i,G} \in I\!R$$

The relationship between $\dot{s}$ and u_c is here characterized only by the first lines of $\mathbf{L}_{\mathrm{S},v}$ and $\mathbf{L}_{\mathrm{S},\omega}$ in (4). By neglecting δ with respect to the point depths, and assuming that the centroid stays 'near' the image plane center, we assume that:

$$\frac{1}{n}\sum_{j=1}^{n}\frac{X_j}{z_j} \ll 1 \qquad \frac{1}{n}\sum_{j=1}^{n}\frac{\delta}{z_j} + X_j^2 \ll 1$$

which leads to $\mathbf{L}_{\mathrm{S},v} = 0$ and $\mathbf{L}_{\mathrm{S},\omega} = 1$. Replacing in (3) leads to the same control law used in [10]:

$$\omega = -\lambda e$$

where λ is a given positive gain. With this method, no metrical knowledge of the 3-D scene is needed, since the controller relies uniquely on the image features. However, we will show that not taking into account 3D information, is a limitation for this control strategy.

VI. SIMULATIONS AND EXPERIMENTS

The simulations and experiments have been carried out on a MagellanPro robot. This is a differential-drive robot with a caster wheel added for stability. The on-board camera is a 30 Hz Sony EVI-D31, with a resolution of 640×480 pixels. For preliminary simulations, we have made use of Webots[3], where a simulated robot with the same kinematic and sensorial characteristics as MagellanPro has been designed. Video clips of the experiments are available at: `www.dis.uniroma1.it/~labrob/research/VisNav.html`.

In Webots, the six controllers have been compared by replaying a taught path of approximately 4.8 m composed of $N = 17$ reference images; we set $\tau_\epsilon = 4$. Using the Webots GPS sensor, we can derive the 3D paths tracked by the simulated robot in the 6 cases (full curves in Fig. 2). Note that, although with all controllers the robot is able to reach the final goal image I_{17}, path tracking is less accurate with RH and AIJCA than with the 4 other controllers. This result is confirmed by the metrics reported in Table I: both the image error ϵ_i with respect to I_i, and the position error with respect to q_i, averaged over the 17 reference images/configurations, are higher for RH and AIJCA, than for the other controllers. The smaller value of the third metric (average number of matched points n on each image) for RH and AIJCA, is both a cause and an effect of lower accuracy: less points provide less information for control, while, inprecise path tracking worsens feature tracking. Although the performances of the 4 other controllers are comparable, slightly better results are obtained when the depth is estimated using 3D reconstruction (IJP and IJC), than when it is fixed (IJPU and IJCU). The importance of the – mainly longitudinal – position error in RH and AIJCA is due to the fundamental role played by the point depths (which are not used by the latter controllers) in the pose accuracy associated to an image-based task: with RH and AIJCA, the robot stops much after configuration q_{17}. To further investigate the controller performances, we have plotted, in Fig. 3 (left), the typical evolution of ω and ϵ_i during a path step. Here, we focus on the step from I_6 to I_7, although the trends are similar for the other 16 steps. The curves show that with RH, which is merely position-based, the value of ω is strongly conditioned by the 3D

[3] www.cyberbotics.com

TABLE I

CONTROLLERS PERFORMANCE IN WEBOTS - CORRECT CALIBRATION

controller	RH	IJP	IJPU	IJC	IJCU	AIJCA
average ϵ_i w.r.t. I_i (pixels)	2.9	1.9	2.4	2.2	2.6	3.1
average position error (cm)	14	4	5	4	6	21
average n	77	94	92	93	92	73

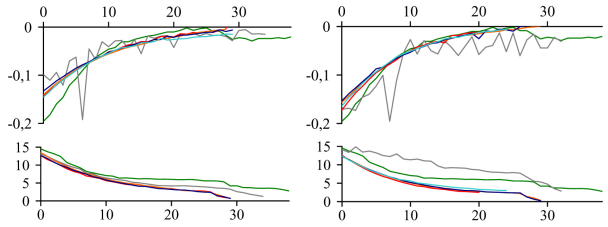

Fig. 3. Evolution of ω (top, in rad/s) and ϵ_7 (bottom, in pixels) at successive iterations while the simulated robot moves from I_6 to I_7 using: RH (grey), IJP (red), IJPU (orange), IJC (blue), IJCU (cyan), and AIJCA (green), with correct (left) and coarse (right) camera calibration.

reconstruction error, and oscillates, leading to later convergence of ϵ_i (hence to the late robot stop). The inaccuracy in $\Delta\theta$ ($\pm 6^\circ$ average estimation error over a total heading variation, throughout the path, of -110°) is due to our choice of estimating on-line the camera pose by using only pairs of images, instead of performing a computationally costly global bundle adjustment. This is consistent with the choice of processing the same sensor input for all controllers, i.e., simply data from the current and from the next reference images. Late convergence of ϵ_i also occurs with AIJCA (green in Fig. 3). Smoother curves are obtained with the other 4 image jacobian controllers, which take into account the feature point image positions, as well as their 3D depths.

To verify the controllers' robustness, the 6 simulations have been repeated with a random calibration error of either $+10\%$ or -10% on each of the camera parameters: f_X, f_Y, δ. For the uniform depth controllers (IJPU and IJCU), we have also included a random calibration error of $+10\%$ or -10% on $\bar{z}$, simulating imprecise tuning of this parameter. For the coarsely calibrated simulations, the replayed paths are represented by the dashed curves in Fig. 2, while the relevant metrics, and the evolution of ω and ϵ_i during the seventh step are shown respectively in Table II, and Fig. 3 (right). For AIJCA, which is independent from the camera parameters, the results are identical to those of the calibrated case. Fig. 2 shows that the robot is able to successfully follow the path in all 6 cases, although path tracking is obviously less precise than in the calibrated camera case. Again, the 4 image jacobian controllers that utilize feature depth, outperform RH (where camera parameters are crucial for control) and AIJCA.

To evaluate the effect of the choice of the parameter $\bar{z}$ used by the two uniform depth controllers, we have repeated the calibrated camera simulations by varying the value of $\bar{z}$ for a fixed gain λ. Since in our workspace the feature points are very sparse (with average depth 1.9 m, and standard deviation 1.5 m), the uniform depth assumption is quite strong. Nevertheless, all the simulations that we have run using $\bar{z} \in [0.8, 500]$ m were successful, and provided good performances: average $\epsilon_i < 3.5$, $n > 80$ and position error < 25 cm. In fact, for $\bar{z} \to \infty$, both IJPU and IJCU rely uniquely on image features, since $\mathbf{L}_{\mathrm{S},v} \to 0$, and $\mathbf{L}_{\mathrm{S},\omega}$

TABLE II

CONTROLLERS PERFORMANCE IN WEBOTS - COARSE CALIBRATION

controller	RH	IJP	IJPU	IJC	IJCU	AIJCA
average ϵ_i w.r.t. I_i (pixels)	3.5	2.7	2.7	2.3	2.5	3.1
average position error (cm)	19	7	7	5	9	21
average n	57	94	96	93	90	73

depends only on the image coordinates of the P_j points; hence, in this case, inappropriate tuning of $\bar{z}$ does not worsen the controllers' performance. On the other hand, for $\bar{z} < 0.8$ m, the simulations fail, due to the large modeling error in the choice of $\bar{z}$, which should be closer to the average value 1.9 m. Therefore, the simulations show that IJPU and IJCU are robust to large $\bar{z}$ modeling errors, and that overestimating $\bar{z}$ is preferable.

To assess the convergence domain, in a fourth series of simulations, the 6 controllers have been tested starting from an initial configuration 'distant' from the teaching phase initial configuration. The distance is evaluated by considering the ratio ρ obtained by dividing the initial image error ϵ_1 (with respect to I_1) in the presence of initial pose error, by the initial ϵ_1 in the ideal case (i.e., when replay starts at the teaching initial configuration). For each controller, we assess the convergence domain by verifying the maximum ρ tolerated. For IJP and IJC, a maximum ρ of 4.1 is tolerated (i.e., these controllers converge from an initial view with ϵ_1 4.1 times larger than the initial teaching view). For IJPU and IJCU, $\rho = 2.6$ is tolerated; for AIJCA and RH, respectively $\rho = 2.1$ and $\rho = 1.9$. Clearly, a complete stability analysis would be required to precisely assess the convergence domain. Nevertheless, these simulation results are useful to confirm the properties of the 6 controllers, and show that IJP and IJC can converge even in the presence of a large initial error.

After the simulation results, we ported the navigation framework on the real MagellanPro for further validation. Since the image jacobian points and centroid controllers have behaved similarly in Webots, we have not tested the centroid controllers IJC and IJCU on the real robot. A taught path of approximately 2.0 m, composed of $N = 4$ reference images, has been replayed using the other 4 controllers, with $\tau_\epsilon = 5$. Since the robustness of the image processing algorithms is not crucial in this work, the environment was lightly structured, by adding artificial visual textures. With RH, the experiment failed after having reached image I_2. The reason is the large position error with respect to the taught path, which causes feature point loss. The replayed paths, are shown, along with the taught path (white) in Fig. 4. Values of the main metrics are reported in Table III, and the evolution of ω and ϵ_i while the robot approaches I_1 are shown in Fig. 5. The experiments confirm the controllers' characteristics seen in Webots. Indeed, both the attempt of accomplishing an image-based task by using merely 3D features (RH), and that of tracking accurately the 3D path by using merely image features (AIJCA) are unfruitful, while the two complete image jacobian controllers provide the best performances both in the image and in the 3D state space. Again, IJP, which utilizes computed depths, outperforms IJPU, which utilizes an approximation of the depths. This result is even

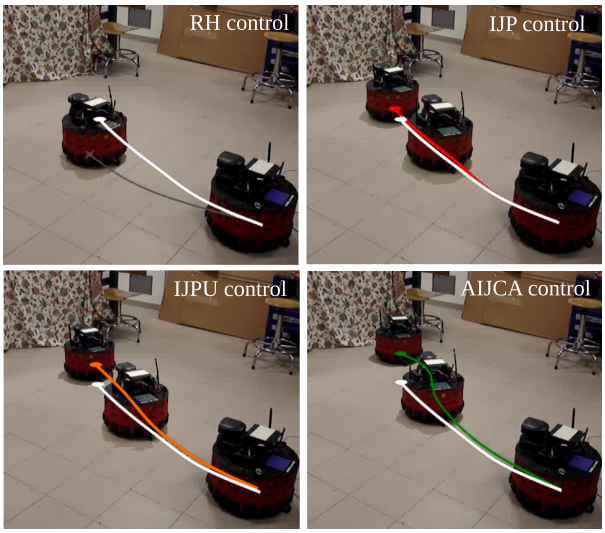

Fig. 4. Replaying a taught path (white) using controllers: RH (grey), IJP (red), IJPU (orange), and AIJCA (green). Robot key positions during navigation are also shown: initial, intermediate and, for the 3 successful controllers, final positions.

more evident in a real environment than in simulations. Fig. 4 also confirms that in all cases, the predominant component of the position error is in the longitudinal direction (i.e., in the z direction), as outlined in the simulations. This is not a surprise, since it is well known that for nonholonomic systems, set-point regulation (which cannot be achieved via smooth time-invariant feedback) is more difficult to achieve than trajectory tracking. Besides, the importance of the longitudinal position error is due to the utilization of scale-dependent Harris points, which are hard to track when the motion in the optical axis direction is important. However, since the object of this study is the control, rather than the sensing technique, this is not crucial: using scale-invariant features will improve navigation, without modifying the controllers' characteristics.

VII. CONCLUSIONS AND FUTURE WORK

We have compared 6 appearance-based controllers for nonholonomic navigation from a visual memory. The simulations and experiments have shown that the 4 complete image jacobian controllers, which combine both image data and feature depth, outperform the 2 controllers which utilize only 3D data, or only image data. Besides, although 3 controllers necessitate 3D reconstruction, for the image jacobian controllers (IJP and IJC), a large 3D reconstruction error (e.g., due to coarse camera calibration) can be allowed without jeopardizing performance. Indeed, in the IJP and IJC experiments, as opposed to the RH experiments, 3D reconstruction performed on-line by using only pairs of subsequent images gave excellent results. Moreover, since 3D reconstruction introduces computational delay at run time, and increases sensitivity to image noise, a valid alternative is to use the uniform depth controllers IJPU and IJCU. We hope that the results of this study can be useful for the researchers working on similar visual navigation frameworks worldwide. Future work will be devoted to taking into account environment modifications between the teaching and replaying phases. We also plan to implement and integrate obstacle avoidance, by considering cases where the robot must deviate from the taught path in order to avoid an obstacle, while maintaining localization accuracy.

TABLE III

COMPARING FOUR CONTROLLERS ON THE REAL ROBOT

controller	RH[a]	IJP	IJPU	AIJCA
average ϵ_i w.r.t. I_i (pixels)	4.2	3.0	3.8	4.5
average position error (cm)[b]	40	30	33	44
average n	42	63	57	32

[a]Since RH failed after I_2, these are averaged over 2 replay steps.
[b]Estimated from the videos of the experiments.

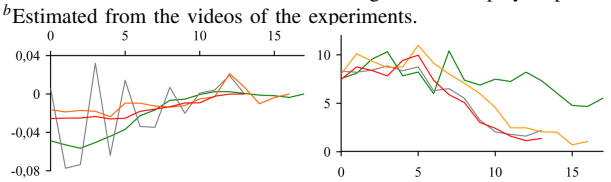

Fig. 5. Evolution of ω (left, in rad/s) and ϵ_1 (right, in pixels) while the robot moves towards I_1 using controllers: RH (grey), IJP (red), IJPU (orange), and AIJCA (green).

REFERENCES

[1] G. N. DeSouza and A. C. Kak, "Vision for Mobile Robot Navigation: a Survey", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, 2002.
[2] F. Chaumette and S. Hutchinson, "Visual servo control tutorial, part I and II", *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, and vol. 14, no. 1, 2007.
[3] Y. Masutani, M. Mikawa, N. Maru and F. Miyazaki, "Visual servoing for non-holonomic mobile robots", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1994.
[4] T. Ohno, A. Ohya and S. Yuta, "Autonomous navigation for mobile robots referring pre-recorded image sequence", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1996.
[5] E. Royer, M. Lhuillier, M. Dhome and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation", *Int. Journal of Computer Vision*, vol. 74, no. 3, pp. 237–260, 2007.
[6] G. Blanc, Y. Mezouar and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes", *IEEE Int. Conf. on Robotics and Automation*, 2005.
[7] O. Booij, B. Terwijn, Z. Zivkovic, B. Kröse, "Navigation using an appearance based topological map", *IEEE Int. Conf. on Robotics and Automation*, 2007.
[8] T. Goedemé, M. Nuttin, T. Tuytelaars and L. Van Gool, "Omnidirectional vision based topological navigation", *Int. Journal of Computer Vision*, vol. 74, no. 3, pp. 219–236, 2007.
[9] Y. Matsumoto, M. Inaba and H. Inoue, "Visual navigation using view-sequenced route representation", *IEEE Int. Conf. on Robotics and Automation*, 1996.
[10] A. Diosi, A. Remazeilles, S. Šegvić and F. Chaumette, "Outdoor Visual Path Following Experiments", *IEEE/RSJ Int. Conf. on Intelligent Robots and System*, 2007.
[11] A. A. Argyros, K. E. Bekris, S. C. Orphanoudakis and L. E. Kavraki, "Robot homing by exploiting panoramic vision", *Autonomous Robots*, vol. 19, no. 1, pp. 7-25, 2005.
[12] S. S. Jones, C. Andersen and J. L. Crowley, "Appearance based processes for visual navigation", *IEEE/RSJ Int. Conf. on Intelligent Robots and System*, 1997.
[13] D. Burschka and G. Hager, "Vision-based control of mobile robots", *IEEE Int. Conf. on Robotics and Automation*, 2001.
[14] C. Harris and M. Stephens, "A combined corner and edge detector", *4th Alvey Vision Conference*, pp. 147–151, 1988.
[15] Y. Ma, S. Soatto, J. Košecká and S. S. Sastry, "An Invitation to 3-D Vision: from Images to Geometric Models". New York: Springer, 2003.
[16] P. A. Beardsley, A. Zisserman and D. W. Murray, "Sequential Updating of Projective and Affine Structure from Motion", *Int. Journal of Computer Vision*, vol. 23, pp. 235–259, 1997.
[17] A. Remazeilles and F. Chaumette, "Image-based robot navigation from an image memory", *Robotics and Autonomous Systems*, vol. 55, no. 4, pp. 345–356, 2007.
[18] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects", *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1116–1127, 2005.