Visual Servoing of a Wheeled Mobile Robot for Intercepting a Moving Object

Francesco Capparella, Luigi Freda, Marco Malagnino, and Giuseppe Oriolo

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy
francesco.capparella@tin.it, {freda,oriolo}@dis.uniroma1.it, m_malagnino@tiscali.it

***Abstract*— We present a vision-based scheme for driving a nonholonomic mobile robot to intercept a moving target. Our method relies on a two-level approach. On the lower level, the pan-tilt platform which carries the on-board camera is controlled so as to keep the target at the center of the image plane. On the higher level, the robot operates under the assumption that the camera system achieves perfect tracking. In particular, the relative position of the ball is retrieved from the pan/tilt angles through simple geometry, and used to compute a control law driving the robot to the target. Various possible choices are discussed for the high-level robot controller. The proposed visual interception method is validated through simulations as well as experiments on the mobile robot MagellanPro.**

***Index Terms*— Visual servoing, interception, nonholonomy.**

I. Introduction

In this paper, we address the problem of intercepting a moving target via a nonholonomic mobile robot through visual feedback. Interception (approaching to a moving object until collision) and tracking (approaching a moving object while matching its location and velocity) are important tasks in a wide range of applications, such as robotic soccer and automated surveillance. Moreover, they represent a challenging testbed for the integration of various techniques involving image processing, filtering, control theory and AI strategies.

Interception and tracking have been largely addressed in the literature with different approaches depending on the target motion characteristics as well as the robot kinematic and dynamic model. The instantaneous target location and velocity can be known in advance as part of a reference trajectory or estimated and predicted trough sensory data.

A general strategy known as PPE (Prediction, Planning and Execution) focuses on time-optimal interception, possible when the target motion is completely (or to a large extent) known in advance; in this case, the problem becomes essentially that of planning a robot trajectory leading to a suitable rendezvous point. For example, a time-optimal technique for free-flying interceptors and targets is given in [1]. This kind of approach, combined with conventional tracking methods, can be extended to manipulators in order to achieve smooth grasping interception with terminal-velocity matching [2]. In [3], an active PPE technique for a 6-dof manipulator has been designed to improve the original PPE strategy.

When the target motion is not known in advance and/or may suddenly change, it is necessary to rely on some sort of estimation of its position and velocity coming from sensor data. A particularly interesting situation is the use of visual feedback [4]. In *position-based* visual servoing, the target posture is estimated on the basis of visual data and geometric models. For instance, an omnidirectional vision system is used in [5] to determine the robot posture from feature extraction, so that the basic tasks of a robotic goalkeeper can be accomplished in terms of trajectory tracking and posture stabilization. In this context, Kalman filtering is often used to obtain robust prediction of the target motion [6], [7], [8].

In *image-based* visual servoing, the spatial relationship between the target and the robot camera is directly estimated on the image plane, and the error signal is expressed in terms of image features. Originally developed for robotic manipulators equipped with eye-in-hand systems, image-based visual servoing has also been applied to nonholonomic vehicles such as wheeled mobile robots [9].

Many works address trajectory tracking directly on the image plane. An image-based motion planning scheme is proposed in [10], where a virtual trajectory is directly generated and tracked on the image plane. In [11], a mobile robot tracks road edges in a panoramic image sequence. In [12], a wheeled vehicle tracks a trajectory represented in terms of image sequences of an object, and the controlled camera motion is used as an input for visual servoing. A similar idea was used in [13]: while the camera tracks the ball on the image plane, pan and tilt angles are used as a visual input for a probabilistic motion control scheme.

Finally, we mention that *human-like strategies* (like the so-called LOT and OAC) have been proposed for mobile robots in [14], [15]. In particular, an optical acceleration cancellation heuristic is used for ground balls interception in [16].

Our approach to the problem is to design a two-level interception scheme which makes use of visual information only. On the lower level, the pan-tilt platform which carries the on-board camera is controlled so as to keep the target at the center of the image plane. On the higher level, the robot operates under the assumption that the camera system achieves perfect tracking. In particular, the relative position of the ball is retrieved from the pan/tilt angles through simple

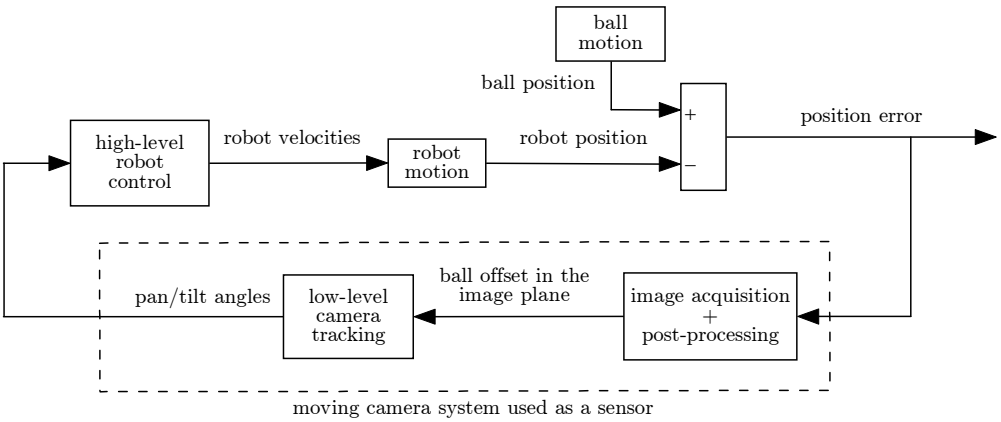

Fig. 1. A block diagram description of the proposed approach

geometry, and used to compute a control law driving the robot to the target. Various possible choices for the high-level robot controller are discussed in the paper. To validate the overall visual interception method, simulations as well as experiments on the mobile robot MagellanPro are presented.

II. General description of the approach

Our objective is to devise a vision-based control method in order to drive a wheeled mobile robot to intercept a moving target. In particular, we shall take the following assumptions:

1) The target and the robot move on the same plane in the absence of obstacles.
2) The moving target is a ball whose trajectory is generic and unknown in advance.
3) The robot is subject to a nonholonomic rolling constraint. In particular, it has the kinematics of a unicycle:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned}$$

where x, y are the Cartesian coordinate of the robot in a fixed frame, θ is its orientation with respect to the same frame, and v, ω are the robot linear and angular velocities.

4) The robot is equipped with an on-board camera mounted on a pan-tilt platform.

The generality of the second assumption is motivated by a possible application to robotic soccer, where the ball changes frequently speed and direction. The consequence of this viewpoint is that, differently from other techniques, our visual servoing approach does not make *any* prediction of the ball movement: all the control laws to be presented use only the ball position and velocity as estimated from visual data.

We propose a two-level solution whose conceptual scheme is outlined in Fig. 1, with reference to a problem setting illustrated in Fig. 2. The basic idea is that the camera system is controlled through the pan-tilt platform so as to keep the ball at the center of the image: this requires post-processing of the acquired image in order to retrieve the coordinates of the ball center w.r.t. the center of the image plane (the *ball offset*), and then a camera tracking controller which moves the pan/tilt platform in such a way that the offset is zeroed. Then, under the assumption of perfect tracking (i.e., zero ball offset), the relative position of the ball with respect to the robot can be retrieved through simple geometry from the pan/tilt angles, and used to drive the robot toward the ball. In practice, the pan/tilt camera system acts as a sensor providing information about the relative motion of the target with respect to the ball.

The above approach is reminiscent of a human-like behavior: the body follows the head while the latter visually pursues a target. The implicit assumption of *separation* between the camera and the robot closed-loop dynamics (very similar to the property underlying observer-based stabilization schemes) appears reasonable also in view of the fact that the pan-tilt platform dynamics is typically much faster than the robot dynamics. Note also that the robot odometric information is never used; as a consequence, the controllers to be presented in the following are purely image-based.

For illustration, our method will be discussed with reference to our MagellanPro robot. This is a differential-drive robot with a caster wheel added for stability; its shape is circular with a diameter of 40 cm. The on-board camera is a Sony EVI-D31, with a refresh time of 0.13 sec and

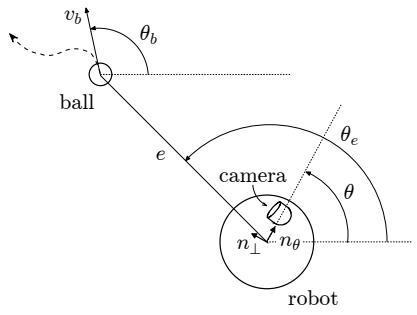

Fig. 2. Problem setting with definition of some relevant variables

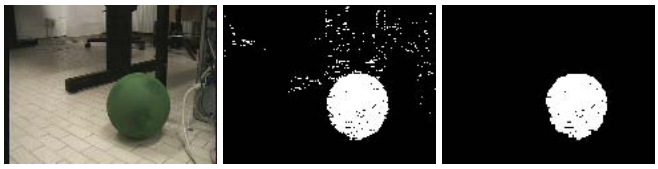

Fig. 3. *Left*: The image acquired by the camera. *Center*: After color recognition. *Right*: After noise filtering.

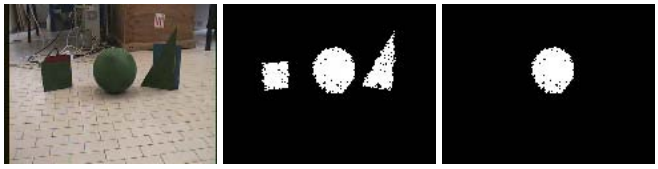

Fig. 4. *Left*: Three objects with the same color. *Center*: After color recognition and filtering. *Right*: After shape recognition.

a resolution of 160x120 pixels. The camera is mounted on a pan-tilt platform which provides two degrees of freedom. For simplicity, we shall neglect the small displacement (about 11 mm) between the pan and the tilt axes.

III. Image processing and camera tracking

The camera tracking algorithm must keep the ball at the center of the image plane. To this end, it is first necessary to recognize the ball in the image on the basis of its known color and shape.

A. Image processing

First, color recognition is used to obtain a binary image. To reduce noise, the image is then filtered with erosion/dilation techniques [17] as shown in Fig. 3.

If in the environments there are other objects with the same color of the ball, shape recognition based on Hu-moments [18] is also performed. The result of shape recognition is shown in Fig. 4.

B. Camera tracking

Denote by $\xi = (\xi_x \ \ \xi_y)^T$ the ball offset (i.e., the vector of the ball coordinates w.r.t. the center of the image), computed from the image as the centroid of the ball. For the design of the camera tracking method, we make the simplifying assumption that, for a given position of the ball, ξ_x and ξ_y in the image plane depend respectively on the pan angle φ and on the tilt angle φ only. Under such assumption, the expression of these relationships can be computed by planar geometry.

For illustration, consider the dependence of ξ_x on the pan angle φ. Refer to Fig. 5 for the geometric setting and definitions; note that the pan center (i.e., the point where the pan axis intersects the camera axis) is assumed to coincide

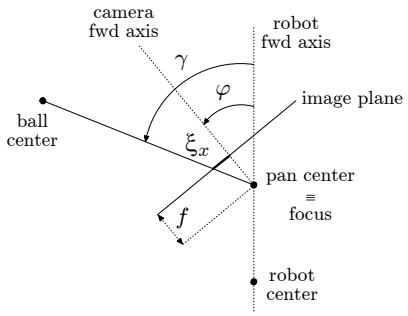

Fig. 5. Dependence of the ball image coordinate ξ_x on the pan angle φ

with the focus of the camera[1]. Being

$$\xi_x = f \tan(\gamma - \varphi),$$

the differential kinematics of ξ_x is

$$\dot{\xi}_x = \frac{f}{\cos^2(\gamma - \varphi)}(\dot{\gamma} - \dot{\varphi}).$$

As customary in visual servoing, we shall design the controller based on the kinematic model, i.e., assuming the pan velocity $\dot{\varphi}$ as an available input. Note that $\dot{\gamma}$ is an exogenous variable because it depends on the instantaneous relative motion between the ball and the robot.

It may be easily shown that choosing the pan velocity as

$$\dot{\varphi} = \dot{\gamma} - k_\varphi \xi_x \qquad k_\varphi > 0$$

guarantees exponential convergence of ξ_x to zero. The feedforward term $\dot{\gamma}$ can be computed numerically with the aid of the following relationship:

$$\gamma = \varphi + \arctan \xi_x / f.$$

The addition of this feedforward term compensates for any relative motion between the ball and the robot, thus increasing the level of separation between the camera and the robot dynamics.

A similar construction in the vertical plane holds for the dependence of the other ball coordinate ξ_y on the tilt angle ψ. Hence, choosing the tilt velocity as

$$\dot{\psi} = \dot{\eta} - k_\psi \xi_y \qquad k_\psi > 0$$

guarantees exponential convergence of ξ_y to zero. Here, η is the angle between the horizontal axis and the line joining the camera focus and the ball center, computed as

$$\eta = \psi + \arctan \xi_y / f.$$

Again, the feedforward $\dot{\eta}$ is computed via finite difference.

Figure 6 shows the performance of the above camera tracking technique in two experiments executed with the MagellanPro robot. In the first experiment, the robot and

[1]One may verify that such a simplification is acceptable in practice provided that the displacement between these two points (less than 3 cm for the MagellanPro) is much smaller than the distance between the ball center and the pan axis.

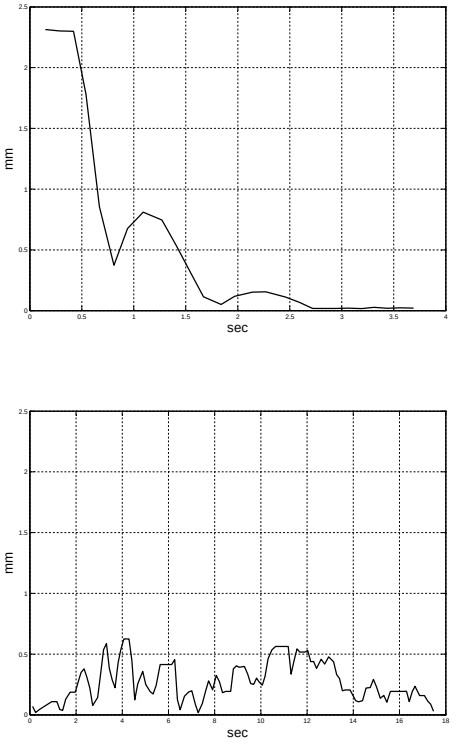

Fig. 6. Experimental results for camera tracking in terms of the norm $|\xi|$ of the ball offset. *Above*: the robot and the ball are motionless. *Below*: the robot is moving along a straight path at a speed of 0.2 m/s.

the ball are motionless; the camera successfully hunts the ball and places it at the center of the image, zeroing the ball offset. In the second experiment, the robot is moving along a straight path at a speed of 0.2 m/s, while the ball is still. The norm of the ball offset $|\xi|$ is very small and essentially due to the quantization introduced by the camera. These satisfactory results justify *a posteriori* the assumption that ξ_x and ξ_y respectively depend only on φ and ψ.

IV. Robot Control

In this section we describe the three different robot controllers that have been tried in our interception scheme. First, the control laws are presented in a *complete information* context: both the system state (the robot position and orientation) and the exosystem state (the ball) are assumed to be known. Then, it is shown how these control law can be implemented knowing only the pan/tilt angles, based on the assumption that the camera tracking system effectively keeps the ball at the center of the image.

A. Structure of the controllers

Refer again to Fig. 2 for the geometry of the problem. Let e be the error vector, v_b the ball velocity vector, and n_θ, $n_\perp$ the unit vectors respectively directed as and orthogonal to the robot forward axis. In the following, we will refer to $\theta_e - \theta$ as the *pointing-error* and to $\theta_b - \theta$ as the *alignment error*: the former represents the angle between the robot forward axis and the line of sight to the target; the latter is the angle between the instantaneous directions of motion of the target and the ball.

The first controller, inspired to a Cartesian position regulator [19], is expressed as

$$\begin{aligned} v &= k_1\, e \cdot n_\theta + v_b \cdot n_\theta \\ \omega &= k_2(\theta_e - \theta) + \dot{\theta}_e \end{aligned} \tag{1}$$

The linear velocity is composed by a feedback and a feedforward term, respectively obtained by projecting on the forward robot axis the error vector and the ball velocity vector. The angular velocity includes a feedback action proportional to the pointing error and a feedforward term given by the derivative of the angle θ_e. Although details are omitted here, a Lyapunov-like analysis allows to establish global convergence of the unicycle robot to the ball under the above control law (see [20]).

A second controller may be derived from the first by including in the angular velocity the feedback term $e \cdot n_\perp$, i.e. the projection of the error vector along the orthogonal direction to the forward axis of the robot

$$\begin{aligned} v &= k_1\, e \cdot n_\theta + v_b \cdot n_\theta \\ \omega &= k_2(\theta_e - \theta) + k_3\, e \cdot n_\perp + \dot{\theta}_e \end{aligned} \tag{2}$$

This additional feedback term forces a faster alignment of the robot with the line of sight of the target when the norm of the vector error is large, i.e., when the robot is far from the ball. This should result in a more effective interception of the target. Also for this controller, convergence of the robot to the ball can be proven by Lyapunov arguments [20].

The third control law chosen for our experimental study is a well-known trajectory tracking method [21] for nonholonomic targets. This controller is quite similar to the previous ones, the difference being the use in the angular velocity of the alignment error $\theta_b - \theta$ in place of the pointing error $\theta_e - \theta$ and of a nonlinear gain for the second feedback term;

$$\begin{aligned} v &= k_1\, e \cdot n_\theta + v_b \cdot n_\theta \\ \omega &= k_2(\theta_b - \theta) + k_3 |v_b| \tfrac{\sin(\theta_b - \theta)}{\theta_b - \theta}\, e \cdot n_\perp + \dot{\theta}_b \end{aligned} \tag{3}$$

Note that, consistently, the feedforward term in ω is different as well. With respect to the previous schemes, controller (3) aims more at tracking than at intercepting the target.

It is worth to note that eqs. (1–3) prescribe the same linear velocity for the robot. One basic difference among these controllers is the following: while the latter requires that the target does not stop, controllers (1–2) will still guarantee interception in that case, due to their nature of position (as opposed to posture) stabilizers.

B. Implementation of the controllers

Under the assumption of perfect camera tracking (i.e., the ball is always kept at the center of the image plane), all the relevant variables needed to implement the abovementioned controllers can be computed by the pan/tilt angles. In the following, we give explicit formulas for implementing eq. (1);

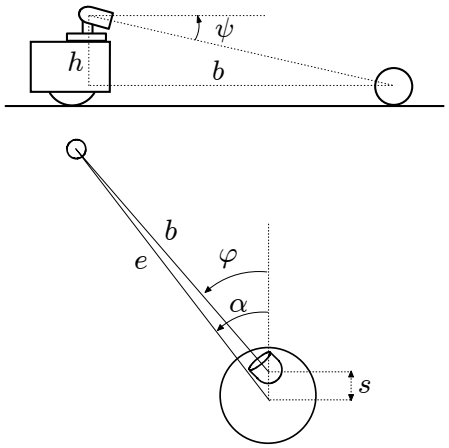

Fig. 7. Determination of the ball position with respect to the robot from the pan/tilt angles. *Above*: Side view. *Below*: Top view.

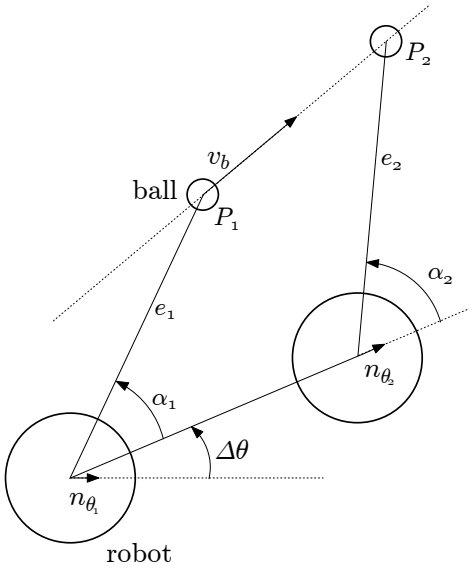

Fig. 8. Estimation of the ball velocity

similar derivations, omitted here, hold for the other two control laws [20]. Refer to Figs. 7 and 8 for the relevant geometry and the definition of auxiliary variables.

Denoting with b and h respectively the horizontal and vertical displacement between the tilt axis and the ball center, we have:

$$b = \frac{h}{\tan\psi},$$

where ψ is the tilt angle. Also, the application of Carnot's theorem gives

$$|e|^2 = b^2 + s^2 - 2bs\cos(180^\circ - \varphi),$$

where s is the horizontal displacement between the robot center and the pan axis, while α is the angle between the error vector and the forward axis of the robot, computed as

$$\alpha = \arcsin\frac{b\sin(180^\circ - \varphi)}{|e|}.$$

Hence, the feedback term for computing the linear velocity of controller (1) is obtained as

$$e \cdot n_\theta = |e|\cos\alpha,$$

while the pointing error for computing the angular velocity is directly given by

$$\theta_e - \theta = \alpha.$$

The feedforward term in the linear velocity of controller (1) can be approximated through finite difference as follows. Figure 8 shows the ball in two positions P_1 and P_2 attained at consecutive instants t_1 and t_2 separated by a sampling interval T. An easy calculation yields

$$v_b \cdot n_\theta \approx v_1\cos\frac{\omega_1 T}{2} + \frac{|e_2|\cos\alpha_2}{T} - \frac{|e_1|}{T}\cos(\alpha_1 - \omega_1 T),$$

where the subscript i ($i = 1, 2$) refers to the time instant in which the quantity is defined. Note in particular that v_1, ω_1 are known because they are the linear and angular velocity inputs to the robot at the previous sampling instant. Similarly, for the feedforward term in the angular velocity we have

$$\dot\theta_e \approx \omega_1 + \frac{\alpha_2 - \alpha_1}{T}.$$

V. Simulations and Experiments

Before proceeding to an experimental validation of our method on the MagellanPro robot, a simulation study was carried out to assess its performance.

First, a comparison of the various controllers discussed in Sect. IV-A was made within MATLAB, assuming that all the relevant variables and their derivatives were directly available. A typical result is shown in Fig. 9, where the robot and the ball are respectively represented as a triangle and a circle. The ball is moving on along a line with a constant speed of 2 m/s. The plots a)–f) are obtained as follows (all the control gains were set to a unit value):

a) controller (1) with $\dot\theta_e = 0$;
b) controller (1);
c) controller (2) with $\dot\theta_e = 0$;
d) controller (2);
e) controller (3) with a constant gain for the second feedback term (this is the result of a linear control design, see [21]);
f) controller (3).

The omission of the feedforward term $\dot\theta_e$ in plots a) and c), motivated by the desire to reduce the real-time computational load of the control law, results in a less efficient interception with respect to plots b) and d), where the robot is more effectively driven to the ball. As for plot e) and f), the full nonlinear controller (3) performs better than its linearized version. On the average, these two schemes show a marked tracking (as opposed to interception) attitude consistently with the use of the alignment error $\theta_b - \theta$ in place of the pointing error $\theta_e - \theta$.

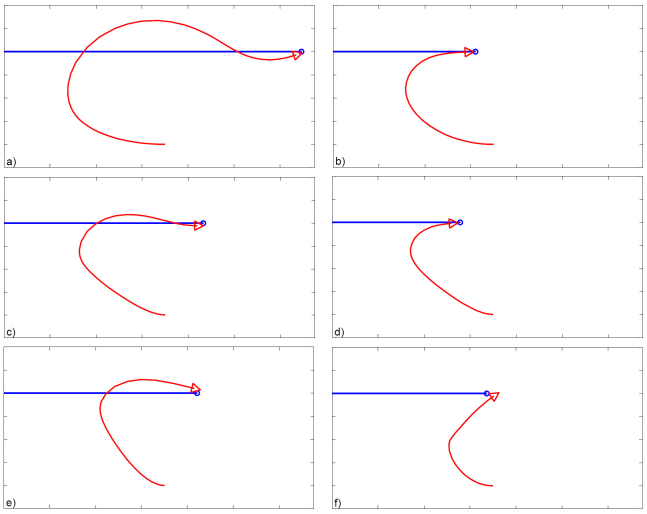

Fig. 9. MATLAB simulation: robot trajectories obtained applying the various control laws.

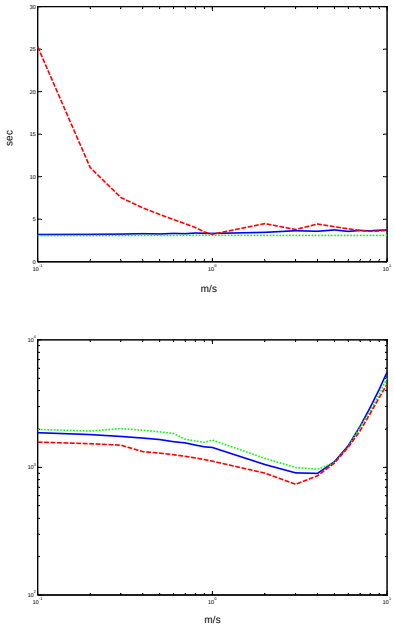

Fig. 10. Results obtained for a ball velocity/control gain ratio between 1 and 10. *Above*: Time needed for interception. *Below*: Control effort. Continuous blue line: controller (1). Green dotted line: controller (2). Red dashed line: controller (3)

Figure 10 summarizes the results obtained with controllers (1–3) varying between 1 and 10 the ratio of the ball velocity to the control gain, in terms of time needed for interception and of control effort (integral of the squared velocity norm). While controller (3) was found to be slightly less demanding in terms of control effort, it also needed a long time for intercepting slow balls, at least for this particular choice of control gains. Interestingly, the interception time of controllers (1) and (2) was approximately constant over the ball velocity range.

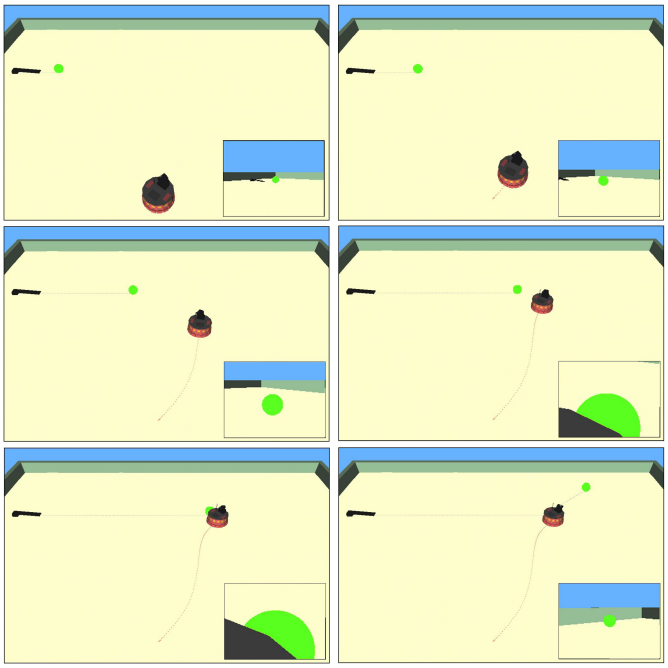

Fig. 11. Simulation of an interception in Webots

To perform a more realistic simulation, accounting also for the presence of the pan-tilt camera system, we implemented our visual control scheme using Webots[2]. The simulated camera, which reproduces exactly the characteristics (including noise) of the EVI-D31 mounted on MagellanPro, acquires scene images which are then processed using the techniques presented in Sect. III in order to recognize the ball and keep it at the center of the image. The various control laws are then visually implemented as explained in Sect. IV-B.

A simulation result obtained with the control law (2) is shown in Fig. 11. The camera views, visible in the bottom-right corner of each snapshot, show that the ball is not perfectly centered with respect to the image, and even partially occluded by the robot own body at close range. Nonetheless, interception is successfully completed, confirming that the assumed separation holds and that the proposed two-level scheme is quite robust with respect to these nonidealities.

Finally, we present some experimental results obtained with the actual MagellanPro using the control law (2). Figure 12 illustrates a successful interception experiment by means of superimposed snapshots. In this case, the ball was traveling along an approximately linear path.

A more challenging experiment is shown in Fig. 13. Here, the ball was passed back and forth by two human players in a manner reminiscent of a common soccer training exercise. The robot does a good job in hunting and finally intercepting the ball in spite of its frequent motion reversals. Movie clips of these and other experiments are available at the web page http://www.dis.uniroma1.it/labrob/research/VisualInterception.html.

[2]Webots is a robot simulation environment developed by Cyberbotics Ltd. (http://www.cyberbotics.com).

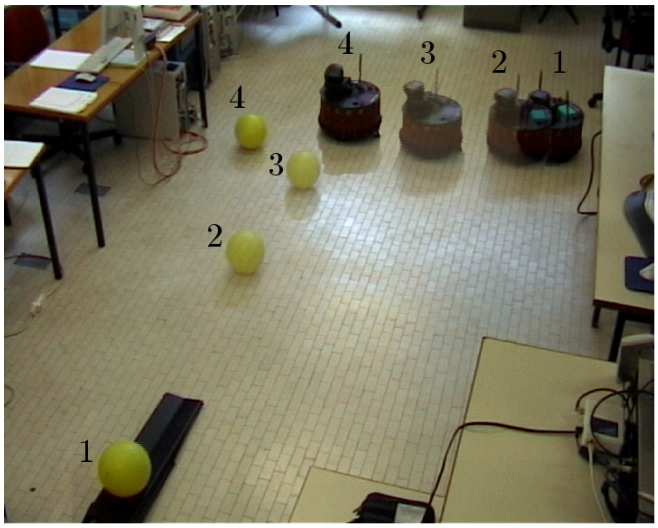

Fig. 12. A successful interception experiment with the ball traveling on a linear path. The numbers identify the position of the ball and the robot at four different time instants.

Fig. 13. An experiment where the robot tracks a ball passed back and forth by two human players

VI. Conclusions

We have presented a two-level vision-based scheme for driving a nonholonomic mobile robot to intercept a moving target. On the lower level, the pan-tilt platform carrying the on-board camera is controlled so as to keep the target at the center of the image plane. On the higher level, the robot operates under the assumption that the camera system achieves perfect tracking. In particular, the relative position of the ball is retrieved from the pan/tilt angles through simple geometry, and used to compute a control law driving the robot to the target. Various possible choices have been discussed for the high-level robot controller. The proposed visual interception method has been validated through simulations as well as experiments on the mobile robot MagellanPro.

Current work includes the addition of a Kalman filter to provide robust estimates of the ball position and velocity, the use of instantaneous odometric data (instead of past control inputs) in the visual computation of the quantities needed by the high-level robot controller, and a campaign of comparative experiments aimed at assessing the relative performance of our method with respect to other interception schemes.

References

[1] C.-D. Yang, F.-B. Hsiao, and F.-B. Yeh, "Generalized guidance law for homing missiles," *1989 IEEE Trans. on Aerospace and Electronic Systems*, vol. 25, pp. 197–212, 1989.

[2] M. Mehrandezh, N. M. Sela, R. G. Fenton, and B. Benhabib, "Robotic interception of moving objects using an augmented ideal proportional navigation guidance technique," *2000 IEEE Trans. on Systems, Man and Cybernetics, Part A*, vol. 30, pp. 238–250, 2000.

[3] D. Hujic, E. A. Croft, G. Zak, R. Fenton, J. K. Mills,and B. Benhabib, "The robotic interception of moving objects in industrial settings: Strategy development and experiment," *1998 IEEE Trans. on Mechatronics*, vol. 3, pp. 225–239, 1998.

[4] P. I. Corke, Visual Control of Robots: High Performance Visual Servoing, RSP Press, 1996.

[5] H. Lausen, J. Nielsen, M. Nielsen, and P. Lima, "Model and behavior-based robotic goalkeeper," *RoboCup 2003 International Symposium*, 2003.

[6] G. Artus, P. Morin, and C. Samson, "Tracking of an omnidirectional target with a nonholonomic mobile robot," *2003 ICAR Int. Conf. on Advanced Robotics*, pp. 1468–1473, 2003.

[7] A. K. Das, R. Fierro, V. Kumar, B. Southall, J. Spletzer, and C. J. Taylor, "Real-time vision-based control of a nonholonomic mobile robot," *2001 IEEE Int. Conf. on Robotics and Automation*, pp. 1714–1719, 2001.

[8] C. H. Messom, G. S. Gupta, S. Demidenko, and L. Y. Siong, "Improving predictive control of a mobile robot: Application of image processing and kalman filtering," *IMTC Instrumentation and Measurement Technology Conference*, 2003.

[9] D. P. Tsakiris, P. Rives, and C. Samson, "Applying visual servoing techniques to control nonholonomic mobile robots," *1997 IROS Int. Conf. on Intelligent Robots and Systems*, 1997.

[10] H. Zhang and J. P. Ostrowski, "Visual motion planning for mobile robots," *2002 IEEE Trans. on Robotics and Automation*, pp. 199–208, 2002.

[11] P. I. Corke, D. Symeonidis, and K. Usher, "Tracking road edges in the panospheric image plane," *2003 IEEE/SRJ Int. Conf. on Intelligent Robots and Systems*, pp. 1330–1335, 2003.

[12] Y. Masutani, M. Mikawa, N. Maru, and F. Miyazaki, "Visual servoing for non-holonomic mobile robots," *1994 IROS Int. Conf. on Intelligent Robots and Systems*, pp. 1133–1140, 1994.

[13] C. Coué and P.Bessière, "Chasing an elusive target with a mobile robot," *2001 IEEE Int. Conf. on Robotics and Automation*,pp. 1370–1375, 2001.

[14] J. A. Borgstadt and N. J. Ferrier, "Interception of a projectile using a human vision-based strategy," *2000 IEEE Int. Conf. on Robotics and Automation*,pp. 3189–3196, 2000.

[15] R. Mori and F. Miyazaki, "GAG(Gaining Angle of Gaze) strategy for ball tracking and catching task," *2002 IEEE/RSJ International Conference on Robots and Systems*, pp. 281–286, 2002.

[16] K. Mu, T. G. Sugar, and M. K. McBeath, "Perceptual navigation strategy: A unified approach to interception of ground balls and fly balls," *2003 IEEE Int. Conf. on Robotics and Automation*, pp. 3461 - 3466, 2003.

[17] J. C. Russ, The Image Processing Handbook, CRC Press, 2002.

[18] M. K. Hu, "Visual Pattern Recognition By Moment Invariants," IRE Trans. on Information Theory, vol. 8, pp. 179–187, 1962.

[19] A. De Luca and G. Oriolo, "Local incremental planning for non-holonomic mobile robots," *1994 IEEE Int. Conf. on Robotics and Automation*, pp. 104-110, 1994.

[20] F. Capparella, L. Freda, M. Malagnino, and G. Oriolo, "Visual servoing of a wheeled mobile robot for intercepting a moving object," DIS Report, Università di Roma "La Sapienza", 2005.

[21] C. Samson, "Time-varying feedback stabilization of car-like wheeled mobile robots," *International Journal of Robotics Research*, vol. 12, no. 1, pp. 55–64, 1993.