IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)
October 29 - November 3, 2001. Hawaii, USA.

# Chasing an Elusive Target With a Mobile Robot

Christophe Coué, Pierre Bessière
GRAVIR - INRIA RHONES-ALPES
ZIRST - 655, av de l'europe.
38330 Montbonnot Saint Martin.
France.
tel: (+33) 4 76 61 54 36
fax: (+33) 4 76 61 52 10
Christophe.Coue@inrialpes.fr
Pierre.Bessiere@imag.fr

## Abstract

This paper describes how a mobile robot (a six-wheeled Koala equipped with a PAL pan-tilt camera) can chase an elusive target (a remote controlled toy car) in a unknown and unconstrained environment. The first purpose of the paper is to demonstrate efficiency, simplicity, and adequacy of *Bayesian Robot Programming* (BRP) to quickly develop such applications. The second purpose of the paper is to illustrate that tremendous information compression ratio may be obtained by some pertinent sensori-motor decoupling.

**Keywords** — mobile robotic, bayesian robot programming, visual servoing.

# Chasing an Elusive Target With a Mobile Robot

**Christophe Coué,**
INRIA* Rhônes-Alpes - GRAVIR†
ZIRST - 655 avenue de l'Europe.
38330 Montbonnot. France
Christophe.Coue@inrialpes.fr

**Pierre Bessière**
Laplace group, Leibniz lab
46, avenue Félix Viallet
38000 Grenoble. France
Pierre.Bessiere@imag.fr

## Abstract

This paper describes how a mobile robot (a six-wheeled Koala equipped with a PAL pan-tilt camera) can chase an elusive target (a remote controlled toy car) in a unknown and unconstrained environment. The first purpose of the paper is to demonstrate efficiency, simplicity, and adequacy of *Bayesian Robot Programming* (BRP) to quickly develop such applications. The second purpose of the paper is to illustrate that tremendous information compression ratio may be obtained by some pertinent sensori-motor decoupling.

## 1 Introduction

This article is concerned with the chase of an elusive target by a mobile robot equipped with a vision system. By chase, we precisely mean following and catching up a moving target.

A central problem of robotic programming is how to use an incomplete and uncertain model of the environment to perceive, infer, decide, and act efficiently. An original robot programming method that specifically addresses this problem has been proposed by O. Lebeltel in his Ph.D. Thesis [6] and summarized in [7]. This method called *Bayesian robot programming* (BRP) is used in the present article to perform the chase using neither geometric model of the robot nor camera calibration.

The control of a mobile robot based on visual data has to face an essential problem of data reduction. How to deal with the considerable data flow coming from the camera? In this paper we propose to summarize the huge visual data flow by the single head (the camera) pan and tilt. Head and body control are decoupled. On the one hand, we control the head to visually pursue the target. On the other hand, we control the body to pursue the head. Both results are then combined to obtain the desired chasing behavior.

Tsakiris *et al* have shown that it is possible to extend visual servoing techniques to nonholonomic systems by properly adding degrees-of-freedom to the platform, in the form of a hand-eye system [12]. Dias *et al* [1] have addressed the problem of simulating pursuit with a mobile robot and artificial vision. Their solution deals with the interaction of multiple independent processes controlling different degrees of freedom of the vision system and the mobile robot position and orientation. Virtually all existing approaches to visual servoing are based on control theory.

After briefly presenting BRP in the next Section and the experimental platform in Section 3, we illustrate the BRP in Section 4 by addressing the problem of visual servoing of the head of the robot. Section 5 is dedicated to the control of the mobile platform using only the direction and the orientation of the head. Finally, Section 6 illustrates behavior combination by mixing the chase with a reactive behavior of obstacle avoidance.

## 2 Bayesian Robot Programming

We assume that any model of a real phenomenon is incomplete. There are always some hidden variables, not taken into account in the model, that influence the phenomenon. Furthermore, perception and control are inherently uncertain. Uncertainty arises from sensor limitations or noise. Rational reasoning with incomplete and uncertain information is quite a challenge for artificial systems. Bayesian Inference and Learning addresses this challenge [4] relying upon a well established formal theory.

The probabilistic approach has led to fielded systems with unprecedented levels of autonomy and robustness. In recent years it has become the dominant paradigm in a wide array of robotic problems, such as map building, localization and planification [2, 8, 11, 5]. Lebeltel, with

*Inst. Nat. de Recherche en Informatique et en Automatique
†Lab. d'informatique GRAphique, VIsion et Robotique de Grenoble.

the BRP method, proposed a quite different approach based on the concept of *bayesian programs*. Beside mobile robotics, BRP has also been used for arm control and CAD modeling [9].

The usual notion of *logical proposition* (either true or false) is the first key concept of probabilistic reasoning. Logical operators can be used to derive new propositions (conjunction, disjunction, negation). *Discrete variable* is the second concept that is needed: it is a set of logical proposition that are exhaustive and mutually exclusive (at least one is true, only one is true). Discrete variables can be combined too (conjunction). To deal with uncertainty, *probabilities* are attached to propositions, and to manipulate probabilities, usual inference rules are used:

- Conjunction rule:
  $P(X\ Y) = P(X)P(Y \mid X) = P(Y)P(X \mid Y)$.
- Normalization rule: $\sum_X P(X) = 1$.

Where $X$ and $Y$ denote discrete variables and $P$ a probability.

$$Program \begin{cases} Description & \begin{cases} Relevant\ Variables \\ Decomposition \\ Parametric\ Forms \end{cases} \\ Question \end{cases}$$

Fig. 1: Structure of a Bayesian Program.

In this framework, a *Bayesian Program* consists of two parts: a *description* and a *question* (See Fig. 1).

A description is a model of interaction between the robot and its environment. It is a probability distribution defined by:

- the set of *relevant variables* on which the joint distribution is defined (typically motor, sensory and internal variables),
- the *decomposition* of the joint distribution as a product of simpler terms. This terms describes dependent relationship between variables,
- the *parametric forms* assigned to each of the terms appearing in the decomposition.

This model could be used to control the robot. Orders are drawn at random according to distributions called *questions*. A question is obtained by partitioning the set of variables into three sets: the searched variables (typically motor variables), the known variables (tipycally sensor variables) and the free variables (tipycally internal variables). A question on the description $D$ may be expressed as the distribution:

$$P_D(Searched \mid Known). \tag{1}$$

Knowing the joint distribution (the description), It is always possible to compute any possible question (any partition of the variables). To do so, the following general inference is used:

$$\begin{aligned} & P_D(Shearched \mid Known) \\ &= \sum_{Free} P(Searched\ Free \mid Known) \\ &= \frac{\sum_{Free} P(Searched\ Free\ Known)}{P(Known)} \\ &= \frac{\sum_{Free} P(Searched\ Free\ Known)}{\sum_{Searched,Free} P(Searched\ Free\ Known)} \\ &= \frac{1}{Z} \times \sum_{Free} P(Searched\ Free\ Known) \end{aligned}$$

where $Z$ is a normalization term.

Our research group[1] has developed an API called OpenPL which is very close to mathematical language, to express bayesian programs. An inference engine has been implemented to automate bayesian inference. A short description of this engine can be found in [7]. It operates in two stages: a symbolic simplification stage that permits to reduce the complexity of the probability distribution to be computed, and a numeric stage that actually computes the distribution.

## 3 Experimental Platform

Koala is a compact ($30 \times 30$ cm) mobile robot designed at the EPFL[2] and commercialized by K-Team[3]. It carries a PAL pan-tilt camera independently controlled

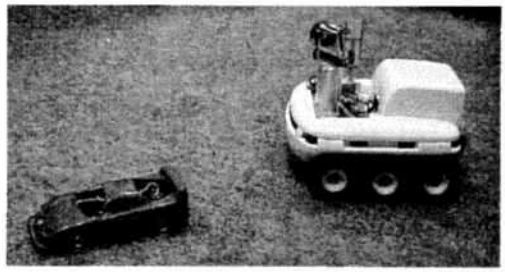

Fig. 2: Koala robot and its target.

[1] http://www-leibniz.imag.fr/LAPLACE/
[2] Ecole Polytechnique Fédérale de Lausanne (Switzerland)
[3] http://www.k-team.com/

and is equipped with a belt of 16 infrared proximeters used for obstacle detection (see Fig. 2).

The camera is controlled in direction and orientation, with values stored in variables $Pan$ and $Tilt$. The robot is controlled by its translation speed $V_{trans}$ and its rotation speed $V_{rot}$.

The target we want the robot to chase is the red radio-controlled car.

## 4 Head Servoing

### 4.1 Object Detection

We are only interested in the position of the gravity center of the target in the image. We use the statistical method described in [10], based on histograms of luminance normalized color. The position of the gravity center of the target in the image is stored in the variables $X$ (horizontal position) and $Y$ (vertical position).

This algorithm is simple and efficient. Furthermore, it is robust to partial target occlusion. As it is a probabilistic algorithm, it has been easily implemented using OpenPL.

### 4.2 Head Servoing

The goal of head visual servoing is to keep the visual target centered in the image. The classical visual servoing approach gives a way to compute differential changes in the image features parameters knowing differential changes in the position of the camera, using the *image Jacobian* [3]. Alternatively, we propose to specify this relationship as a description called $D_{head}$. In the classical approach you have to inverse the jacobian to control the camera. With the BRP approach, this inversion is implemented as a question. In the following, we consider that the two axis of the camera are independent and present only the description for the pan axis.

Two *variables* are relevant here: the motor variable $Pan$, and the sensor variable $X$.

As we want to describe the movement of the target in the image knowing a movement of the camera, we choose the following *decomposition* of the joint distribution:

$$P_{D_{head}}(X\ Pan) = P_{D_{head}}(Pan) P_{D_{head}}(X|Pan). \qquad (2)$$

To complete the description $D_{head}$, we finally need to assign *parametric forms* to each of the terms appearing in (2). We have no *a priori* information about the movement of the camera. Hence $P_{D_{head}}(Pan)$ is a uniform distribution. Depending on the situation, the observation for $X$ may be more or less certain. This is summerized by assigning a Gaussian parametric form $G_{\mu(Pan),\sigma(Pan)}(X)$ to $P_{D_{head}}(X \mid Pan)$. Values of $\mu(Pan)$ and $\sigma(Pan)$ could be learnt by experiment or a priori fixed by the programmer using the following intuitive idea: when the camera moves to the left, the target moves to the right of the image.

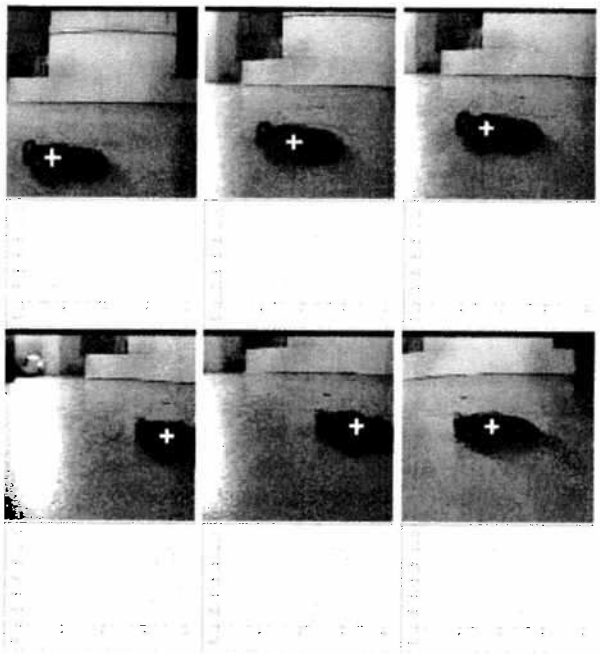

Fig. 3: Evolution of visual servoing of the head when the target is moving. White cross on the car is its center of gravity given by the vision algorithm. Curves represent corresponding $P_{D_{head}}(\ Pan \mid X)$ distributions.

The goal of this program is to control the camera knowing the position of the target in the image. Therefore, the *question* is:

$$P_{D_{head}}(Pan \mid X). \qquad (3)$$

This "inverse" probability distribution is computed every tenth of a second. A value for $Pan$ is drawn at random according to this distribution and sent to the actuators of the camera.

### 4.3 Experimental Results

Figure 3 presents six images of the process. The base of the robot is immobile. Curves under the images represent the probability distribution of the question $P_{D_{head}}(Pan \mid X)$. First the target is immobile and placed on the left of the robot. Algorithm quickly centers the target by moving the camera to the left (first line). Then the target violently accelerates and stops.

We can conclude that detection of the target is accurate enough when the target is the only red object of the scene. Bayesian inference is efficient to solve such an inverse problem, so the camera can be controlled at a frequency of 10 Hz.

This sequence also shows that the large velocity of the camera (up to 400 degrees per second) allows quick movements of the target.

## 5 Body Control For The Chase

### 5.1 Bayesian Program

We want the robot to chase the target: follow it and catch it up. The main idea is to consider only the position of the head to control the body of the robot. In this way the visual information ($100 \times 100 \times 32 = 300$ KBytes) of the target position is summarize as the head position and orientation (2 floats). The corresponding *description* is called $D_{chase}$.

Four *variables* are relevant in this problem: the two motor variables $V_{rot}$ and $V_{trans}$, the camera direction $Pan$ and the camera orientation $Tilt$.

The *decomposition* is chosen using the following assumptions:

- the rotation ($V_{rot}$) and translation ($V_{trans}$) speed of the robot can be controlled independently,
- the rotation speed ($V_{rot}$) of the robot only depends on the head direction ($Pan$),
- the translation speed of the robot depends on the head direction ($Pan$) and orientation ($Tilt$).

We obtain:

$$\begin{aligned} &P_{D_{chase}}(Pan\ Tilt\ V_{rot}\ V_{trans}) = \\ &\quad P_{D_{chase}}(Pan\ Tilt) P_{D_{chase}}(V_{rot} \mid Pan) \\ &\quad P_{D_{chase}}(V_{trans} \mid Pan\ Tilt). \end{aligned} \tag{4}$$

To complete the description $D_{chase}$, we have to assign *parametric forms* to each of the terms appearing in (4).

We have no a priori idea of the target position. Hence $P_{D_{chase}}(Pan\ Tilt)$ is a uniform distribution. We choose a Gaussian parametric form $G_{\mu_1(Pan),\sigma_1(Pan)}(V_{rot})$ to represent the distribution $P_{D_{chase}}(V_{rot} \mid Pan)$. The intuitive idea for computing $\mu_1(Pan)$ is that the more the head is turned to the left, the quicker the body should turn to the left. We choose another Gaussian parametric form $G_{\mu_2(Pan\ Tilt),\sigma_2(Pan\ Tilt)}(V_{trans})$ to represent the distribution $P_{D_{chase}}(V_{trans} \mid Pan\ Tilt)$. The intuitive idea is the closer the robot is from the target, the more the head is looking down. $Pan$ is used to moderate the translation speed of the robot when the target is on its left or on its right

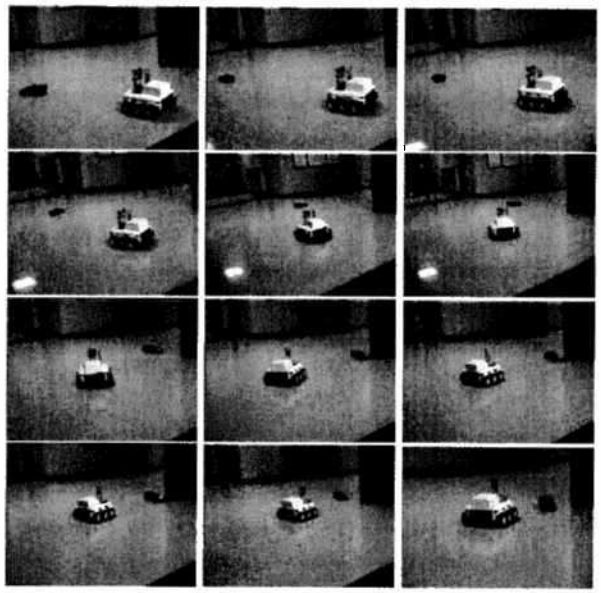

Fig. 4: Example of a chase with the entire system. The video is available at http://www-leibniz.imag.fr/LAPLACE/Films/.

Finally the *question* to control the body of the robot is:

$$P_{D_{chase}}(V_{rot}\ V_{trans} \mid Pan\ Tilt). \tag{5}$$

Like for head servoing, this probability distribution is computed every tenth of a second and the corresponding drawn values of $V_{rot}$ and $V_{trans}$ sent to the motors of the robot.

### 5.2 Experimental Results

Figure 4 shows evolution of the tracking with the entire system. It illustrates the fact that the velocity of the camera allows to track the car even if the body is slower than it.

## 6 Chasing In The Presence Of Obstacles

In section 5, the environment was assumed to be free of obstacles. Now we want the robot to chase its target despite the presence of obstacles. We assume that obstacle can't hide the target from the robot.

### 6.1 Bayesian Program

We dispose of another reactive behavior, obstacle avoidance, based on the proximeters response, and

which control the body on the same variables $V_{rot}$ and $V_{trans}$. This behavior is given by the question $P_{D_{avoid}}(V_{trans}\ V_{rot} \mid Prox\ Dir)$ in which $Prox$ and $Dir$ represent the proximity and the direction of the closest obstacle, and $D_{avoid}$ the corresponding description.

To chase the target despite the presence of obstacles we mix this behavior with the chase of an object, which is given by the question $P_{D_{chase}}(V_{trans}\ V_{rot} \mid Pan\ Tilt\ D_{chase})$ defined in the previous section.

We define a new description $D_{mix}$.

Relevant variables are those defined in the descriptions $D_{avoid}$ and $D_{chase}$. To mix the two behaviors, we introduce a new variable $H$, which acts as a command to switch from avoidance to pursuit. Consequently $H$ can take two values: $a$ or $c$. Finally we consider the seven variables: $\{H, Pan, Tilt, Prox, Dir, V_{rot}, V_{trans}\}$.

We choose the decomposition:

$$\begin{aligned} &P_{D_{mix}}(H\ Pan\ Tilt\ Prox\ Dir\ V_{rot}\ V_{trans}) = \\ &\quad P_{D_{mix}}(Tilt\ Pan) \\ &\quad P_{D_{mix}}(Prox\ Dir) \\ &\quad P_{D_{mix}}(H \mid Prox\ Dir\ Tilt) \\ &\quad P_{D_{mix}}(V_{trans}\ V_{rot} \mid H\ Prox\ Dir\ Tilt\ Pan). \end{aligned} \tag{6}$$

To complete the description $D_{chase}$, we have to assign *paramatric forms* to each of the terms appearing in (6).

We have no idea of target and obstacle positions, consequently $P_{D_{mix}}(Tilt\ Pan)$ and $P_{D_{mix}}(Prox\ Dir)$ are uniform distributions. $P_{D_{mix}}(H \mid Prox\ Dir\ Tilt)$ is a priori defined as a function of $Tilt$, $Prox$, and $Dir$, using the intuituve idea: when the robot is far from any obstacle or close to a lateral one, the probability to do chase is high. When the robot is close to a frontal obstacle, the probability to do avoidance is high. $Tilt$ is used to distinguish the target from an obstacle.

Finally we specify the probability distribution on $V_{trans}$ and $V_{rot}$ as dependent on the value of $H$. In OpenPL, a parametric form could be specified as a question to another description. We use this ability to relate $D_{mix}$ to $D_{chase}$ and $D_{avoid}$.

$$\begin{aligned} &P_{D_{mix}}(V_{trans}\ V_{rot} \mid [H=a]\ Prox\ Dir\ Tilt\ Pan) = \\ &\quad P_{D_{avoid}}(V_{trans}\ V_{rot} \mid Prox\ Dir). \end{aligned} \tag{7}$$

$$\begin{aligned} &P_{D_{mix}}(V_{trans}\ V_{rot} \mid [H=c]\ Prox\ Dir\ Tilt\ Pan) = \\ &\quad P_{D_{chase}}(V_{trans}\ V_{rot} \mid Tilt\ Pan). \end{aligned} \tag{8}$$

Then the question to control the robot is:

$$P_{D_{mix}}(V_{rot}\ V_{trans} \mid Pan\ Tilt\ Prox\ Dir). \tag{9}$$

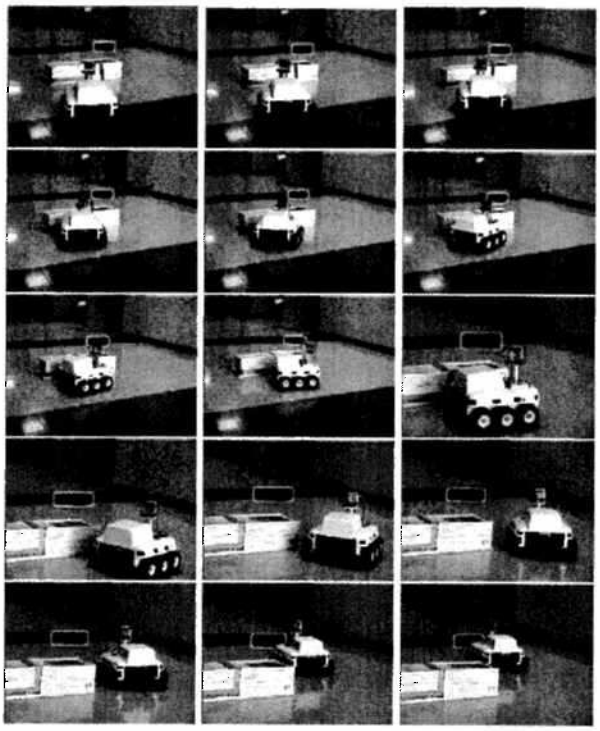

Fig. 5: Example of obstacle avoidance during the chase (the target is inside the white rectangle). The video is available at http://www-leibniz.imag.fr/LAPLACE/Films/.

Result of bayesian inference on the probability distribution expressed in 9 is a sum on the values taken by the variable $H$. Consequently, the robot does not only switch from avoidance to chase, but does a weighted combination of the two behaviors, depending on the sensor values.

### 6.2 Experimental Results

Figure 5 shows an example of chase in which the target is behind an obstacle but still in the field of view of the robot. During the obstacle avoidance, the head of the robot keeps focused on the target, so that the chase can go on after the obstacle has been avoid.

As the method is totaly behavioral, we can't avoid local minima. But as the decision process is stochastic (orders are drawn at random according to a probability distribution), we can hope that the robot goes out "little" local minima.

This experiment shows that BRP allows easy, clear and rigorous specifications of such behavior combination. This seems to be an important benefit compared to some other methods that have great difficulties in

mixing behaviors with one another.

## 7 Conclusion

We presented an application of chase task using an autonomous mobile robot equipped with a vision system. This application has been developed using the Bayesian Robot Programming (BRP) methodology and the associated API OpenPL. The corresponding bayesian programs are very short and simple. They essentially amount to the mathematical equations given in this paper. We show that complexity of control of the body is considerably reduced using internal information such as the direction and orientation of the head. Furthermore, independent control of the head and the body allows easy behavior combination.

Future developments will address more complex sensori-motor devices. We are working on applications of these techniques to both car sensor fusion and control and to complex manipulations tasks with multi arms set up.

## References


[1] J. Dias, C.Paredes, I. Fonseca, H. Araùjo, J. Baptista, and A. Almeida. Simulating pursuit with machine: Experiments with robots and artificial vision. In *IEEE Int. Conf on Robotics and Automation*, volume 1, 1998.

[2] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, 1989.

[3] S. Hutchinson, G. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5), October 1996.

[4] E.T. Jaynes. Probability theorie - the logic of science. unfinished books available at http://bayes.wustl.edu/etj/prob.html, 1998.

[5] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101, 1998.

[6] O. Lebeltel. *Programmation Bayésienne des Robots.* PhD thesis, INPG, Grenoble, France, 1999.

[7] O. Lebeltel, P. Bessière, J. Diard, and E. Mazer. Bayesian robots programming. Technical Report 1, Les cahiers du laboratoire Leibniz, 2000.

[8] J. Leonard and H. Durrant-Whyte. Dynamic map building for an autonomous robot. *Int. Journal of Robotics Research*, 11(4), 1992.

[9] K. Mekhnacha, P. Bessière, and E. Mazer. The design and implementation of a bayesian cad modeler for robotic applications. *Advanced Robotics: the Int. J. of the Robotics Society of Japan*, 15(1), 2001.

[10] K. Schwerdt and J. Crowley. Robust face tracking using color. In *4th International Conference on Automatic Face and Gesture Recognition*, 2000.

[11] S. Thrun, W. Burgard, and D. Fox. A probalistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 31(5), 1998.

[12] D. Tsakiris, P. Rives, and C. Samson. Applying visual servoing techniques to control nonholonomic mobile robots. In *Workshop on New Trends in Image-based Robot Servoing, IROS'97*, 1997.