Positioning a Camera With Respect to Planar Objects of Unknown Shape by Coupling 2-D Visual Servoing and 3-D Estimations

Christophe Collewet and François Chaumette

***Abstract*—This paper proposes a way to achieve positioning tasks by 2-D visual servoing when the desired image of the observed object cannot be precisely described. The object is assumed to be planar and motionless, but no knowledge about its shape or pose is required. First, we treat the case of a threadlike object, and then we show how our approach can be generalized to an object with three particular points. The control law is based on the use of 2-D visual servoing, and on an estimation of two 3-D parameters. We show that this control scheme is not sensitive to the calibration of the camera. We conclude this paper by experimental results relative to objects of unknown shape. In addition, an algorithm to estimate the depth between the object and the camera is provided, which finally leads to a 3-D estimation of the object shape.**

***Index Terms*—2-D visual servoing, 3-D reconstruction, complex images, points of interest, positioning tasks.**

I. Introduction

VISUAL SERVOING is now a classical technique in robot control (see [1] for a description of the different approaches). Nevertheless, in the most often encountered case of an "eye-in-hand" system [2]–[6], we still cannot achieve positioning tasks with regard to partially known objects. Indeed, except rigid manufactured goods for which a model often exists, we rarely have a precise description of the object or of the desired visual features, either because these objects can be subject to deformations or simply because of their natural variability. Such cases appear when we have to treat applications, for example, in the surgical domain, agrifood industry, agriculture, or in unknown environments (underwater, space). That is why we propose, in this paper, a way to achieve accurate positioning tasks by visual servoing related to the objects mentioned above.

In the case of 3-D visual servoing, two approaches exist. The first, and the most often used, is based on the computation of the pose between the object and the camera, requiring a model of the object [7], [8]. Therefore, this approach cannot be used in our case. The second approach is based on 3-D reconstruction by dynamic vision [9]–[11]. This method uses the displacement of the camera and the 2-D motion computed from the image sequence, but it still does not provide accurate results related to the reconstruction errors. On the other hand, active vision [12]–[19] includes strategies to select the motion of the camera to limit such errors. In particular, [12], [13], [16], [19] have shown the benefit of using a *fixation point* (the projection of such a point is centered and motionless in the image during the motion of the camera). However, these approaches have only been used on simple objects. Moreover, let us point out that these 3-D reconstruction techniques are sensitive to the calibration of the system, since the camera displacement or velocity is required.

On the other hand, a lot of works in 2-D visual servoing have shown that the closed-loop system has little sensitivity to calibration errors [20]–[27]. Similar results have been obtained concerning the recent 2-1/2-D approach [28]. However, the 2-D approach, as well as the 2-1/2-D, cannot cope with the studied objects not being precisely described. Indeed, let us consider an accurate positioning task related to such objects, a raw ham, for example. First, the shape of such objects depends on the way they have been placed by the operator on their support. Second, even though these objects are part of the same class, they are different enough from each other not to permit use of generic visual features for all the objects in the class. In such applications, the desired visual features have to be considered as unknown. Few authors relate such cases. In [29], the authors use a specific motion to achieve an alignment task without a precise description of the desired visual features. Unfortunately, their study is restricted to planar motions. In [30], thanks to dynamic visual features, a positioning task consisting of moving the camera to the position parallel to a planar object of unknown shape is achieved. However, such an approach needs particular motion parameters estimation [31] currently leading to a high computation duration and, consequently, to a low control scheme rate. Moreover, this approach is not well suited to positioning tasks since motion is always required.

The method described in this paper is based on the use of points as visual features. It can achieve positioning tasks with any specified orientation and for the same objects as those used in [30], planar and motionless objects of unknown shape. The proposed method can be seen as an active vision scheme to perform accurate positioning tasks. More precisely, it is based on

Manuscript received March 22, 2001; revised November 23, 2001. This paper was recommended for publication by Associate Editor P. Allen and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by The Brittany County Council (Région Bretagne). This paper was presented in part at the IEEE International Conference on Robotics and Automation (ICRA '01), Seoul, Korea, May 21–26, 2001.

C. Collewet is with Cemagref Rennes, 35044 Rennes Cedex, France (e-mail: christophe.collewet@cemagref.fr).

F. Chaumette is with IRISA/INRIA Rennes, 35042 Rennes Cedex, France (e-mail: chaumett@irisa.fr).

Publisher Item Identifier S 1042-296X(02)05948-7.

1042-296X/02$17.00 © 2002 IEEE

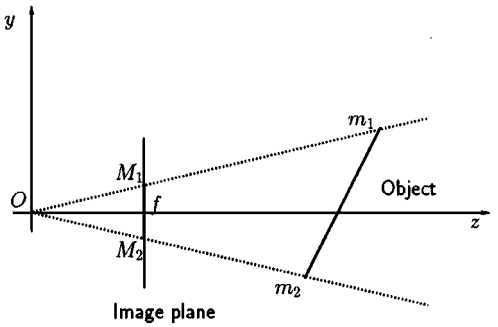

Fig. 1. Projection of the object on the image plane.

the use of 2-D visual servoing and on the estimation of two 3-D parameters which will be specified more precisely below. We will see that this structure uses a fixation point and maintains the object of interest in the field of view of the camera. Moreover, contrary to 3-D approaches, this method does not need a calibrated camera, even though it provides dimensions of the object and an estimation of its range. Unfortunately, it is quite sensitive to the calibration of the robot, since it necessitates a measurement of the camera displacement. Finally, our method combines, in part, the advantages of both 2-D visual servoing and 3-D reconstruction.

This paper is organized as follows. In Section II, we deal with a threadlike object to achieve the particular positioning task consisting of moving the robot to the position parallel to this object. Next, we show in Section III how this approach can be readily generalized for an object with three particular point features and for any specified orientation of the camera. Experimental results are given in the case of a simple object. Moreover, we consider the case of a coarse calibrated camera. Experimental results concerning objects of unknown shape are finally presented in Section IV.

II. Case of a Simple Object: The Segment

A. Task Specification

In this section, we propose to move a robot parallel to a threadlike object by visual servoing. In order to implement applications such as those described in the previous section, we consider the length of the object, as well as its pose, unknown. The object is described by a segment $[m_1, m_2]$ with $\underline{Om_1} = (x_1, y_1, z_1)^T$ and $\underline{Om_2} = (x_2, y_2, z_2)^T$, expressed in the camera frame which is centered in the optical center O. The optical axis of the camera is the z axis of the camera frame (see Fig. 1).

This object projects on the image plane by a perspective projection as a segment $[M_1, M_2]$ of length l with $\underline{OM_1} = (X_1, Y_1, f)^T$ and $\underline{OM_2} = (X_2, Y_2, f)^T$ according to

$$\underline{OM} = \frac{f}{z}\underline{Om} \tag{1}$$

where f is the focal length. Following, without loss of generality, f is assumed to be equal to 1.

We say that the task is achieved, i.e., the camera is parallel to the object, when the optical axis is orthogonal to $[m_1, m_2]$ and the image centered with respect to the y axis. Therefore, we have $Y_1 = -Y_2$ and $z_1 = z_2 = z^*$.

Even though the object is very simple, it is not easy to achieve the task, since an infinite number of object poses leads to a centered image without achieving the task. Nevertheless, a way to achieve this is to center the object in the image, and then to scan other positions while keeping the image centered. We will see that proceeding in this way leads to the use of a fixation point and keeps the object in the field of view of the camera. To do this, we propose to use a method very similar to the redundancy framework of the *task function* approach [32]. So, we will consider as the *main task* the one which maintains the image centered, and the *secondary task* the one which scans other positions while ensuring the object remains centered in the image. These two tasks constitute the so-called *positioning task*. We will see, at the end of this section, that prior to this task, a *preliminary task* will have to be achieved. However, first we briefly present some fundamentals and notations of 2-D visual servoing.

B. 2-D Image-Based Visual Approach

In image-based visual servoing, the control scheme is performed on the basis of visual features extracted directly from the image. From a vector $\underline{s}$ describing the current visual features, the goal is to move the robot so that $\underline{s} = \underline{s}^*$ where $\underline{s}^*$ describes the features when the robot is at its desired position. Such an approach needs knowledge of the interaction matrix (or Jacobian) which describes the relationship between the camera velocity T_c and the velocity of the visual features $\dot{\underline{s}}$

$$\dot{\underline{s}} = L_{\underline{s}}^T T_c \tag{2}$$

where $T_c = (\underline{V}^T, \underline{\Omega}^T)^T$ with $\underline{V} = (v_x, v_y, v_z)^T$ and $\underline{\Omega} = (\omega_x, \omega_y, \omega_z)^T$ are the translational and rotational components of T_c, respectively.

A vision-based task $\underline{e}$ can then be defined when $L_{\underline{s}}^T$ is full rank by

$$\underline{e} = \widehat{L_{\underline{s}}^{T^+}} (\underline{s} - \underline{s}^*) \tag{3}$$

where $\widehat{L_{\underline{s}}^T}$ is a model or an approximation of $L_{\underline{s}}^T$.

In the case of a motionless object, the camera velocity can be obtained simply by [2]

$$T_c = -\lambda \underline{e} \tag{4}$$

with λ being a positive gain.

C. Description of the Positioning Task

1) Main Task: A way to center the segment in the image is to choose as visual features the vector

$$\underline{s} = (X_1, X_2, Y_1 + Y_2)^T \tag{5}$$

and as the desired visual features the vector

$$\underline{s}^* = (0, 0, 0)^T. \tag{6}$$

We immediately obtain the interaction matrix at the desired position [1], [2]

$$L_{\underline{s}^*}^T = \begin{pmatrix} -\frac{1}{\widehat{z^*}} & 0 & 0 & 0 & -1 & Y_1 \\ -\frac{1}{\widehat{z^*}} & 0 & 0 & 0 & -1 & -Y_1 \\ 0 & -\frac{2}{\widehat{z^*}} & 0 & 2\left(1+Y_1^2\right) & 0 & 0 \end{pmatrix} \quad (7)$$

where $\widehat{z^*}$ is an approximation of z^*.

2) Secondary Task: At this step, we want to equate z_1 to z_2, while ensuring that the image of the object is centered. To achieve this, the corresponding motion has to belong to $\mathrm{Ker} L_{\underline{s}^*}^T$ to ensure $\underline{s} = \underline{s}^*$. Therefore, we can apply a control law of the following form to achieve the task

$$T_c = -\lambda_{po} L_{\underline{s}^*}^{T^+} (\underline{s} - \underline{s}^*) + T_s \quad (8)$$

with $T_s \in \mathrm{Ker}\ L_{\underline{s}^*}^T$.

Since $\dim \mathrm{Ker} L_{\underline{s}^*}^T = 3$, several different types of motion exist, and we have chosen that which sets $v_z = \omega_y = \omega_z = 0$ to ensure an x axis rotation. We obtain

$$T_s = (0 \quad \widehat{z^*}\omega_x \left(1+{Y_1}^2\right) \quad 0 \quad \omega_x \quad 0 \quad 0)^T \quad (9)$$

in which ω_x has to be determined.

If $\underline{s} = \underline{s}^*$, only T_s generates a motion. Intuitively, it is easy to see that the length of the segment $l = Y_1 - Y_2$ supplies useful information. In particular, we will show that l takes its maximum value when the camera is parallel to the object. The secondary task is thus equivalent to the maximization of l. In order to prove this, and to obtain all the parameters needed for (8), we show now how l changes during the motion of the camera. Furthermore, we will see that the trajectory of the camera can supply all the unknown 3-D parameters.

D. Modeling of the Length of the Segment

Let us consider the points $m_1 = (y^*, z^*)^T$, $m_2 = (-y^*, z^*)^T$, and $m_0 = (0, z_0)^T$ expressed in the camera frame when the camera is at the desired position. We consider that the motion during the maximization of l results from an x axis rotation $\mathcal{R}$ of angle θ_x centered in m_0 and a translation $\mathcal{T}$ (see Fig. 2). Therefore, θ_x is the orientation error with respect to the desired position. After the motion, m_1 and m_2 can thus be expressed as

$$\begin{cases} m_1' = \mathcal{R}(m_1 - m_0) + \mathcal{T} \\ m_2' = \mathcal{R}(m_2 - m_0) + \mathcal{T}. \end{cases} \quad (10)$$

By perspective projection, we obtain $M_1 = (X_1, Y_1)^T$ and $M_2 = (X_2, Y_2)^T$ (respectively the projection of m_1' and m_2')

$$\begin{cases} X1 = 0 \\ Y1 = -\dfrac{y^* \cos\theta_x + (z^* - z_0)\sin\theta_x}{y^* \sin\theta_x - (z^* - z_0)\cos\theta_x - z_0} \\ X2 = 0 \\ Y2 = -\dfrac{y^* \cos\theta_x - (z^* - z_0)\sin\theta_x}{y^* \sin\theta_x + (z^* - z_0)\cos\theta_x + z_0}. \end{cases} \quad (11)$$

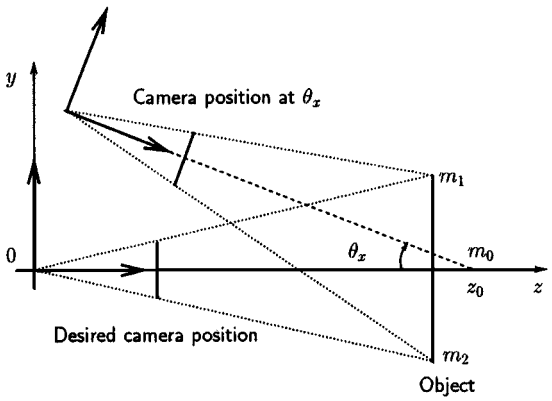

Fig. 2. Modeling of the length of the segment with respect to θ_x.

Then, we search $z_0(\theta_x)$ to keep $Y_1 + Y_2$ at zero, and to perform a motion which belongs to $\mathrm{Ker} L_{\underline{s}^*}^T$ as seen in Section II-C.2. After some tedious manipulations, we obtain [33]

$$z_0(\theta_x) = \frac{z^*}{2} \frac{2c_x - 1 - \sqrt{1 - 4v^2(c_x - 1)c_x}}{c_x - 1} \quad (12)$$

where $v = y^*/z^*$ and $c_x = \cos\theta_x$. We also obtain the modeling of the length of the segment, denoted $l_m(\theta_x)$

$$l_m(\theta_x) = \frac{2v\left(2c_x + 1 - \sqrt{1 - 4v^2(c_x - 1)c_x}\right)}{2v^2(c_x - 1) + 1 + \sqrt{1 - 4v^2(c_x - 1)c_x}}. \quad (13)$$

We can verify, as expected, that $l_m(\theta_x)$ is a 2π periodic and even function. It equates to zero at $\theta_x = \pi/2$ and to $2v$, which is the length of the segment in the image when the camera is parallel to the object, at $\theta_x = 0$. Moreover, it takes a unique maximum at $\theta_x = 0$.

This function, quite a bit complex, can be written more simply for a small value of v by using an approximation at third order

$$l_a(\theta_x) = 2v\cos\theta_x \quad (14)$$

while (12) becomes simply at second order

$$z_0(\theta_x) = z^*. \quad (15)$$

This last relation shows that, when v is small, the control law given by (8) leads to a constant location for m_0. Thus, m_0 is a fixation point.

E. Estimation of the Camera Trajectory

From (12), we can derive the trajectory of the camera expressed in the camera frame (see Fig. 2)

$$\begin{cases} z = z_0(\theta_x)(1 - \cos\theta_x) \\ y = z_0(\theta_x)\sin\theta_x. \end{cases} \quad (16)$$

From this system it is difficult to eliminate θ_x, since z_0 is a nonlinear function of it. Nevertheless, if we consider small values

for v and if we use (15) in (16), the trajectory becomes a circle centered in $(0, z^*)$ with a radius z^*

$$y^2 + (z - z^*)^2 = z^{*2}. \tag{17}$$

Therefore, by using the odometry of the robot, it is possible to extract an approximation $\widehat{z^*}$ of z^* from its trajectory, and, of course, to use it in (8). Furthermore, thanks to the perspective transformation, 3-D features can be obtained as will be shown in Section III-C.2.

F. Validation of the Approximations

We examine in this section how to impose a small value for v, and thus, prove that (14) is a valid approximation. To do this, let us study the behavior of the relative error $E_v^l(\theta_x) = (l_a - l_m)/l_m$ between (13) and (14). After some manipulations, one can show that $E_v^l(\theta_x)$ is a positive and increasing function of v, which zeroes only for $v = 0$. Consequently, for a given approximation error $E_v^{l^*}$, we can find v^* such that $\forall \theta_x \in [-\pi/2, \pi/2]$, $E_v^l(\theta_x) < E_v^{l^*}$. This particular value is given by

$$v^* = 2\sqrt{E_v^{l^*}\left(1 + E_v^{l^*}\right)}. \tag{18}$$

We have now to ensure in practice that $v \leq v^*$, to be sure that the approximation error will be smaller than $E_v^{l^*}$.

If we denote $l^* = 2v^*$, the condition $v \leq v^*$ is equivalent to $l(\theta_x = 0) \leq l^*$ since $l(\theta_x = 0) = 2v$ [see (13)]. However, this constraint is not useful in practice, since it imposes a condition at the end of the positioning task, once the task is achieved (represented by the point B in Fig. 3). It is more interesting to derive a condition at the beginning of this task (point A in Fig. 3). This can be done as follows. By using (14) for a given value $\theta_l{}^* < \pi/2$ and for $v = v^*$, the length of the image at the beginning of the task can be written as $l_{t=0}^* = l^* \cos\theta_l{}^*$. Now, we show that the condition $l_{t=0} \leq l_{t=0}^* \Rightarrow l(\theta_x = 0) \leq l^*$ (with $l_{t=0}$ the length of the image of the segment at the beginning of the positioning task for any orientation error $\theta_x \leq \theta_l{}^*$). Indeed, by substituting $l_{t=0} = l(\theta_x = 0)\cos\theta_x$ and $l_{t=0}^* = l^* \cos\theta_l{}^*$ in the relation $l_{t=0} \leq l_{t=0}^*$, we have

$$l\,(\theta_x = 0) \leq l^* \frac{\cos\theta_l{}^*}{\cos\theta_x} \tag{19}$$

in which the second member is lower than or equal to l^* since $\theta_x \leq \theta_l{}^*$. Consequently, if we impose $l_{t=0} \leq l_{t=0}^*$, we are sure that the modeling given by (14) is valid $\forall \theta_x \in [-\theta_l{}^*, \theta_l{}^*]$, since $E_v^l(\theta_x) \leq E_v^{l^*}$.

In practice, we will see in Section II-H, Section III-C.2, and Section IV that it is possible to choose a high value for $\theta_l{}^*$ and a small value for $E_v^{l^*}$ (typically suitable values are, respectively, 55° and 0.5%).

To really impose a value $l_{t=0}^*$ for l at the beginning of the positioning task, a solution is to add a preliminary task before the positioning task. The most simple way is to choose the following vector as visual features

$$\underline{s}_1 = (X_1, X_2, Y_1, Y_2)^T \tag{20}$$

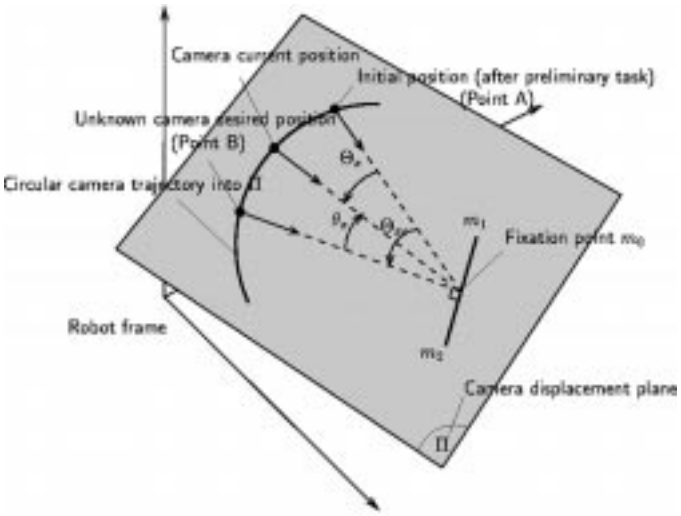

Fig. 3. Camera trajectory during the positioning task.

and as the desired visual features

$$\underline{s}_1^* = \left(0, 0, \frac{l_c}{2}, -\frac{l_c}{2}\right)^T \tag{21}$$

where $l_c = \min(l_{t=0}^*, l_{t=0})$.

Remark: The same study concerning the relative error $E_v^z(\theta_x)$ between (12) and (15) can be led. In other words, we can show that

$$E_v^{z*} \approx 4E_v^{l^*} \tag{22}$$

with E_v^{z*} the value for which $\forall \theta_x \in [-\theta_l{}^*, \theta_l{}^*]$, $E_v^z(\theta_x) < E_v^{z*}$. Thus, if $E_v^{l^*}$ is small enough, the approximation of (12) by (15) is valid, too.

G. Application to Control

Once the preliminary task has been achieved, we have to move the camera to the position parallel to the segment (such that $\theta_x = 0$). However, θ_x is computed from the desired camera position, which is unknown. To alleviate this problem, we can rewrite θ_x as $\theta_x = \Theta_x - \Theta_{xc}$ where Θ_x is the camera angle between its current position and its initial position (at the beginning of the positioning task), and Θ_{xc} is the unknown desired angular value when the camera is parallel to the object (see Fig. 3). This value Θ_{xc} is computed from the length of the image of the segment as follows.

During a motion, the form of which is given by the control law (8), we proceed to an on-line estimation of l_p, the length of the segment in pixels. Since $\underline{s} = \underline{s}^*$, we have $l_p = Y_{p1} - Y_{p2}$. Thereafter, by using the intrinsic parameters, Y_{pi} can be expressed as $Y_c + F_y Y_i (i = 1, 2)$ where Y_c is the coordinate of the principal point in pixels, and $F_y = f/p_y$ (where p_y is the vertical size of a pixel) yielding $l_p = F_y l$, with l modeled by (14). By substituting θ_x by $\Theta_x - \Theta_{xc}$ in (14) we obtain

$$l_p = c_1 \cos\Theta_x + c_2 \sin\Theta_x \tag{23}$$

with

$$\begin{cases} c_1 = 2vF_y \cos\Theta_{xc} \\ c_2 = 2vF_y \sin\Theta_{xc}. \end{cases} \tag{24}$$

Therefore, (23) is a linear expression relative to the parameters c_1 and c_2 that we can obtain from a least-squares algorithm with a forgetting factor η [34]. Following on, (24) provides the value of Θ_{xc}

$$\Theta_{xc} = \arctan \frac{c_2}{c_1}. \tag{25}$$

However, if we want to obtain an accurate value for Θ_{xc}, we have to get enough pairs $(\Theta_x|_k, l_k)$. This can be done by setting ω_x in (9) to a given value ω_0 during t_0. The problem is now to compute t_0. One can proceed as follows: since a least-squares algorithm provides a matrix P_k such that $\tilde{P}_k = \sigma^2 P_k$, with $\tilde{P}_k$ the covariance matrix of the estimation error and σ^2 the variance of the noise, and since the coefficients of P_k decrease when the number of data increases, Θ_{xc} will be supposed accurate enough when all the coefficients of P_k will be smaller than a given threshold ν. We will see in simulation that this value can easily be tuned.

Finally, to ensure an exponential decay of the angular error $\Theta_x - \Theta_{xc}$, we then impose

$$\omega_x = K_x \left(\Theta_x - \Theta_{xc}\right) \tag{26}$$

with K_x being a positive gain. Several approaches can be used to tune this gain, like the one proposed in [35], for example. Here, as soon as Θ_{xc} is known, K_x can be tuned so that the task will be achieved in a given duration t_f (if no saturation occurs). Proceeding this way, by integrating (26), and with simple manipulations, we obtain

$$K_x = \frac{1}{t_f - t_0} \ln \frac{\tilde{\Theta}_{xc}}{\Theta_x|_{t=t_0} - \Theta_{xc}} \tag{27}$$

with $\tilde{\Theta}_{xc}$ the value for which the positioning task is considered as achieved (typically $\tilde{\Theta}_{xc}$ =0.05°).

Remark: Note that Θ_x is obtained thanks to the odometry of the robot, and, therefore, does not depend on the intrinsic parameters of the camera. Of course, it depends on an accurate measurement of the displacement of the camera. Note also that Θ_{xc} does not depend on the intrinsic parameters [see (25)]. We will check this nice property in the section devoted to experimental results.

H. Simulation Results

The simulation system consists of a C program describing the behavior of the relations (1), (8), (23), (25), and (26). To do this, only the three points m_0, m_1, and m_2 are considered. The following constants have been used: $\tilde{\Theta}_{xc} = 0.05°$, $\widehat{z^*} = 80$ cm, $w_{sat} = w_0 = 10°/\text{s}$ (with w_{sat} the value for which the x axis rotation saturates), $t_f = 10$ s. The rate of the control law is $\Delta t = 40$ ms, the dimension of the pixels is $8.3\mu \times 8.3\mu$, the focal length is 12.5 mm, the length of the object is 20 cm. In addition, we have chosen ${E_v^l}^* = 0.5\%$ yielding $v^* = 0.142$, $E_v^{z*} = 2\%$, $l^* = 427.7$ pixels and ${\theta_l}^* = 55°$ yielding $l^*_{t=0} = 245.3$ pixels. ν has been tuned as follows. We looked at the time for which $\widehat{\Theta}_{xc}$ is equal to Θ_{xc}, with a relative accuracy of 10% using a uniform noise of ± 3 pixels on image data. This yielded $\nu = 500$.

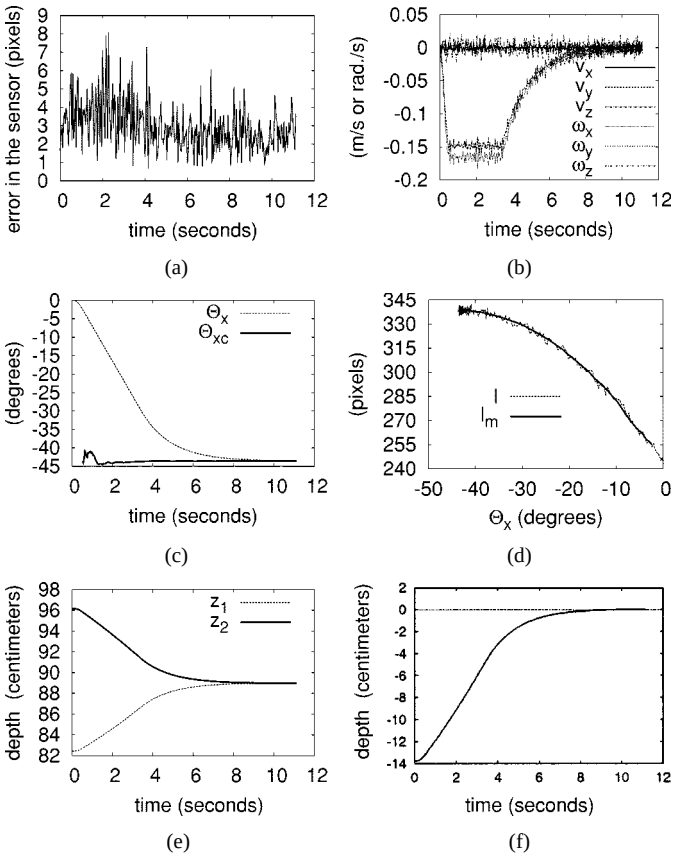

Fig. 4. Simulation results. (a) Error in the sensor. (b) Components of the camera velocity. (c) Desired and current angular values. (d) Measured and modeled image length versus Θ_x. (e) Behavior of z_1 and z_2. (f) Positioning error $z_1 - z_2$.

We depict in Fig. 4 the results obtained from simulation when adding uniform noise (± 2 pixels), and for an orientation error of $-43.5°$. More precisely, Fig. 4(a)–(f) depicts, respectively, the error in the sensor after the realization of the preliminary task (defined by $||\underline{s} - \underline{s}^*||$); the six components of the camera velocity; the desired and current angular values; the measured and modeled image length; the behavior of z_1 and z_2; and the positioning error $z_1 - z_2$. As we can see, in Fig. 4(e) and 4(f), the desired position is reached. Moreover, we can remark in Fig. 4(b) that ω_x is not set immediately to ω_0. A polynomial velocity profile has been used to limit the error in the sensor. Note also that even though t_0 is small (0.52 s), it leads to a low value for the relative error on Θ_{xc} (0.187%). The task is performed in 11.16 s according to the values of t_0 and t_f (10 s). Besides this, we can observe the low noise influence on the positioning error and on the estimation of Θ_{xc}. Thanks to the good modeling of the real situation, we can set the forgetting factor η involved in the estimation of Θ_{xc} to a high value (0.995 in practice, remember that $\eta < 1$) to strongly filter the noise. Moreover, in this example, $l(\theta_x = 0) = 339.6$ pixels yielding $v = l(\theta_x = 0)/2F_y = 0.113$. This low value for v explains why the modeling is good, and why the positioning error is low.

After having treated "segment" objects, we show in the next section how the proposed method can be generalized to a planar object of unknown shape, such that three feature points of the object can be extracted from the image.

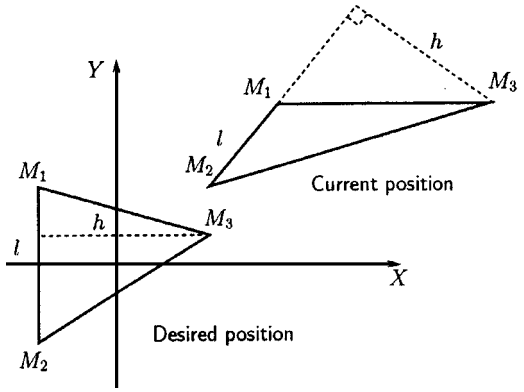

Fig. 5. Triangle built from three feature points of the object.

III. Case of a Planar Object With Three Feature Points

In Section II, the task has been achieved by maximization of the length of the object in the image. In the case of an object described in a 2-D space, a natural transposition consists in maximizing the area S of the object in the image. However, obtaining its analytical expression in the general case seems to be out of reach. Thus, we will simplify the approach by assuming that three feature points of the object can be extracted from the image.

Taking into account three points only, the image becomes very simple as depicted in Fig. 5. A triangle with vertices $M_1(X_1, Y_1)$, $M_2(X_2, Y_2)$, and $M_3(X_3, Y_3)$, which are, respectively, the images of the points $m_1(x_1, y_1, z_1)$, $m_2(x_2, x_2, z_2)$, and $m_3(x_3, y_3, z_3)$.

In these conditions, in the same way we have modeled the length of a segment in Section II-D, we now investigate the area of a triangle.

A. Modeling of the Surface of a Triangle

Let us consider the three feature points m_1, m_2, m_3, and a fixed point m_0, such that, by definition, $m_1 = (-x^*, y^*, z^*)^T$, $m_2 = (-x^*, -y^*, z^*)^T$, $m_3 = (x^*, r^*, z^*)^T$, and $m_0 = (0, 0, z^*)^T$ related to the triangle frame. By perspective transformation, we deduce the coordinates of M_1, M_2, and M_3 at a given position and orientation of the camera. These expressions could lead to the modeling of S. Unfortunately, they are too complex and, consequently, we cannot maximize S directly. Nevertheless, we can maximize sequentially the base l of the triangle and then its altitude h (see Fig. 5), thanks to the modeling of a segment obtained in Section II-D after the preliminary task

$$\begin{cases} l(\theta_x) = Y_1 - Y_2 = 2v\cos\theta_x \\ h(\theta_y) = X_3 - X_1 = 2u\cos\theta_y \end{cases} \tag{28}$$

with $u = x^*/z^*$ and $v = y^*/z^*$.

Consequently, according to the process described in Section II, we specify the following:

1) a preliminary task to ensure that the approximations we will use are valid, as in Section II-F;
2) a positioning task consisting of two tasks, a main task and a secondary task, as in Section II-C. This task will be based on the maximization of S, yielding $z_1 = z_2 = z_3 = z^*$.

Remark: Since we estimate the angular values Θ_{xc} and Θ_{yc} (see Section II-G) for which the camera is parallel to the object, by adding specified constant values, any orientation can be reached as will be shown in Section III-C.2.

B. Control Law

1) Preliminary Task: As mentioned in Section II-F, the goal of the preliminary task is principally to ensure the validation of the modeling obtained in Section II-D when the length of the segment is either l or h.

Due to the symetry of the problem, in the same way we have introduced in Section II-F $E_v^l(\theta_x)$, v^*, l^*, ${\theta_l}^*$, $l^*_{t=0}$, and $l_{t=0}$, we can introduce analogous definitions for $E_u^h(\theta_y)$, u^*, h^*, ${\theta_h}^*$, $h^*_{t=0}$, and $h_{t=0}$ by considering h instead of l.

Consequently, the preliminary task has to ensure $\underline{s}_1 = \underline{s}_1^*$ with $\underline{s}_1$ the visual features given by

$$\underline{s}_1 = (X_1, X_2, X_3, Y_1, Y_2)^T \tag{29}$$

and $\underline{s}_1^*$ the desired visual features given by

$$\underline{s}_1^* = \left(-\frac{h_c}{2}, -\frac{h_c}{2}, \frac{h_c}{2}, \frac{l_c}{2}, -\frac{l_c}{2}\right)^T \tag{30}$$

with $l_c = \min(l^*_{t=0}, l_{t=0})$ and $h_c = \min(h^*_{t=0}, h_{t=0})$.

2) Positioning Task: If we want to maximize l and h, it is more adequate to follow the study described in Section II-C.2, since it yielded to an analytical representation of the length of the segment. Unfortunately, we will see that the vector of the visual features (5) cannot be directly transposed, hence slight modifications have to be taken into account to ensure the same behavior as the control law given by (8). The main problem is to define the correct motions during the maximization of the segments, and thus, the correct visual features to be used to be sure that we can use the results of Sections II-D-G.

Maximizing l imposes $Y_1 + Y_2 = 0$. Similarly, again due to the symmetry of the problem, maximizing h imposes $X_1 + X_3 = 0$. These constraints lead us to define $\underline{s}$ as

$$\underline{s} = (X_1 + X_3, Y_1 + Y_2)^T \tag{31}$$

and the desired visual features as

$$\underline{s}^* = (0, 0)^T. \tag{32}$$

The interaction matrix related to $\underline{s} = \underline{s}^*$ and $z_i = \widehat{z^*}$ ($i = 1, 2, 3$) is given in (33) shown at the bottom of the next page. Hence, the two following vectors which belong to $\mathrm{Ker} L^T_{|\underline{s}^*, \widehat{z^*}}$ could be used to maximize l and h

$$\underline{T}_{sl} = \left(\frac{\widehat{z^*}}{2}\omega_x X_1 (Y_1 - Y_3), \widehat{z^*}\omega_x \left(1 + Y_1^2\right), 0, \omega_x, 0, 0\right)^T \tag{34}$$

$$\underline{T}_{sh} = \left(-\widehat{z^*}\omega_y \left(1 + X_1^2\right), \frac{\widehat{z^*}}{2}\omega_y Y_1 (X_2 - X_1), 0, 0, \omega_y, 0\right)^T. \tag{35}$$

However, these two vectors are different from the one studied in Section II-C.2. Comparing the form of (34) with the one of (9), in (34) a x axis translation appears. We have to impose $Y_1 = Y_3$

to remove the translational component of the x axis of $\underline{T}_{sl}$, and $X_1 = X_2$ for the translational component of y axis of $\underline{T}_{sh}$ to ensure the same behavior as the one studied in Section II-C.2. Unfortunately, these two constraints cannot be satisfied simultaneously in the general case of any triangle. Therefore, we have chosen to impose $X_1 = X_2$, yielding

$$\underline{s} = (X_1 - X_2, X_1 + X_3, Y_1 + Y_2)^T \tag{36}$$

$$\underline{s}^* = (0, 0, 0)^T. \tag{37}$$

The interaction matrix is given in (38), shown at the bottom of the page, and the two vectors $\underline{T}_{sl}$ and $\underline{T}_{sh}$

$$\underline{T}_{sl} = \left(-\widehat{z^*}\omega_x X_1 Y_3 \widehat{z^*}\omega_x \left(1 + X_1^2 + Y_1^2\right),\ 0, \omega_x, 0, -\omega_x X_1\right)^T \tag{39}$$

$$\underline{T}_{sh} = \left(-\widehat{z^*}\omega_y \left(1 + {X_1}^2\right), 0, 0, 0, \omega_y, 0\right)^T. \tag{40}$$

We will see that even if (39) is different from (9), satisfactory results are obtained.

Then, we apply a control law of the form

$$T_c = -\lambda_{po} L^{T^+}_{\underline{|s}^*, \widehat{z^*}} \left(\underline{s} - \underline{s}^*\right) + \underline{T}_{sl} + \underline{T}_{sh} \tag{41}$$

where ω_x and ω_y have to be expressed.

With the control law having been completely defined, we now present the experimental results.

C. Experimental Results

1) Description of the Experimental System: Our experimental system consists of a six-degrees-of-freedom (DOF) robot: three axes of translation and three axes of rotation (concurrent and orthogonal), performed by a "wrist" where a camera (Sony XC-8500CE) is fixed to give an "eye-in-hand" system. An objective with a focal length of 12.5 mm is mounted on this camera. Control and sequencing are performed by a unique PC (Pentium at 200 Mhz under the DOS/4GW operating system), where Matrox Genesis image-processing hardware is connected to a PCI bus, and a Galil DMC-1060 board to an ISA bus to control the robot joints. The code has been written in the C language making use of the 32-b native Genesis library provided by Matrox.

Unless explicity stated otherwise, the value of all the parameters are the same as those used in the simulations (Section II-H).

2) Experimental Results: During the simulations, it was easy to verify how perfect the positioning of the camera was with respect to the object. Experimentally, it is more difficult. To do this, we chose a 3-D pose estimation method [36] which yields the orientation errors θ_x and θ_y. Besides the three points we need, this method requires a fourth one. As specified in [36], it must not belong to the plane defined by our three points. Moreover, we have chosen its coordinates so that once the object and image planes are parallel, it projects onto the principal point of the image. Note that this fourth point is only used for checking and not at all in the control law. In other respects, the rotational part of the pose has been characterized by Euler's angles denoted γ, β, and α, which, respectively, represent the angles of the x, y, and z rotations.

The first experiment consists in positioning the camera parallel to a very simple object composed of three white dots (see Fig. 8). For this experiment, we have chosen the following parameters: $\lambda_{pr} = 1$ (for the preliminary task) and $\lambda_{po} = 8$. All the other parameters are the ones used in simulation. Furthermore, let us point out that we do not compute the variance of the noise, even though it could be done from the residues between the measured and modeled length. We only suppose that it is lower than a given value (9 for all the experiments).

Fig. 6(a)–(d) depicts, respectively, the error in the sensor, the behavior of the components of the camera velocity, the desired and current angular values, and the measured and modeled image length during the maximization of l. Fig. 7 depicts the behavior of the same variables as above, but during the maximization of h. Fig. 8 summarizes the different phases to achieve the positioning task.

First, Figs. 6(d) and 7(d) confirm the theoretical results about the modeling of l and h. Moreover, in Figs. 6(c) and 7(c) we remark that we quickly obtain the desired angular value. The condition $\tilde{P}_k \leq \sigma^2 \nu$ is verified for $t_0 = 0.88$ s for l, and $t_0 = 0.76$ s for h with a relative error on $\Theta_{xc}(\Theta_{yc})$ of 1.81% (2.33%). As expected, the duration of the maximization of l and h is around t_f, respectively, 8.92 s and 9.44 s. On the other hand, let us point out that, as in simulation, the noise does not have a lot of effect on $\widehat{\Theta}_{xc}(\widehat{\Theta}_{yc})$. For this experiment, we have the following.

1) Initial camera pose:
   a) $\gamma = 23.45°$, $\beta = 22.14°$, and $\alpha = 9.64°$;
   b) depth of the feature points (in cm): $z_1 = 75.04$, $z_2 = 70.25$, and $z_3 = 66.55$.

$$L^T_{\underline{|s}^*, \widehat{z^*}} = \begin{pmatrix} -\dfrac{2}{\widehat{z^*}} & 0 & 0 & X_1\left(Y_1 - Y_3\right) & -2\left(1 + X_1^2\right) & Y_1 + Y_3 \\ 0 & -\dfrac{2}{\widehat{z^*}} & 0 & 2\left(1 + Y_1^2\right) & Y_1\left(X_2 - X_1\right) & -X_1 - X_2 \end{pmatrix} \tag{33}$$

$$L^T_{\underline{|s}^*, \widehat{z^*}} = \begin{pmatrix} 0 & 0 & 0 & 2X_1Y_1 & 0 & 2Y_1 \\ -\dfrac{2}{\widehat{z^*}} & 0 & 0 & X_1\left(Y_1 - Y_3\right) & -2\left(1 + X_1^2\right) & Y_1 + Y_3 \\ 0 & -\dfrac{2}{\widehat{z^*}} & 0 & 2\left(1 + Y_1^2\right) & 0 & -2X_1 \end{pmatrix} \tag{38}$$

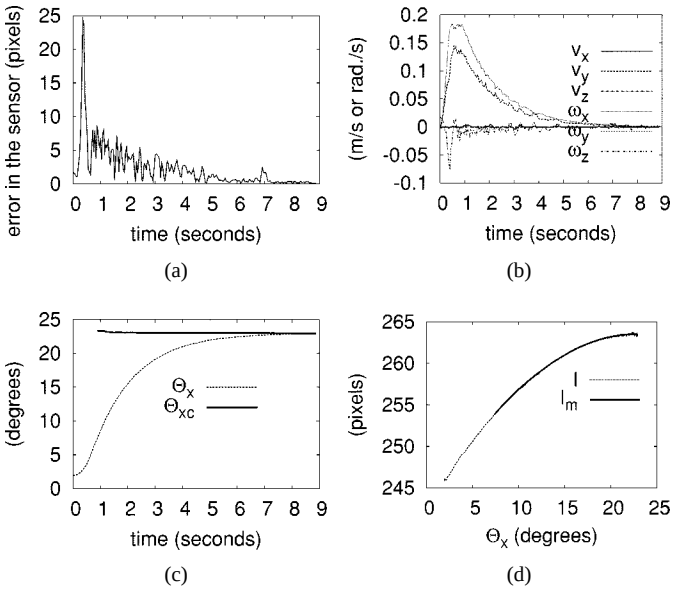

Fig. 6. First experiment: Maximization of l (after the preliminary task). (a) Error in the sensor. (b) Components of the camera velocity. (c) Desired and current angular values. (d) Measured and modeled image length versus Θ_x.

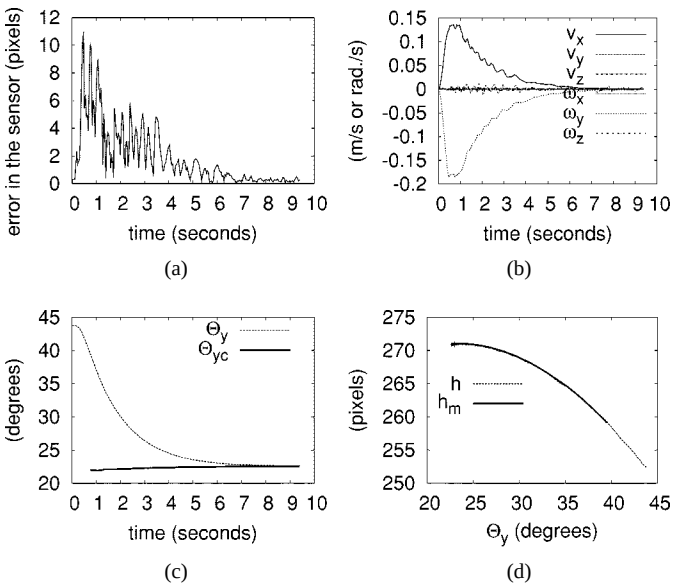

Fig. 7. First experiment—Maximization of h (after the maximization of l). (a) Error in the sensor. (b) Components of the camera velocity. (c) Desired and current angular values. (d) Measured and modeled image length versus Θ_y.

2) Final camera pose:[1]
   a) $\gamma = -0.31°$, $\beta = -0.70°$;
   b) depth of the feature points (in cm): $z_1 = 72.70$, $z_2 = 72.77$, and $z_3 = 72.91$.

If the positioning were perfect, we should have $\gamma = \beta = 0$ and $z_1 = z_2 = z_3$. Therefore, we can conclude that these results are satisfactory. Note that these good results confirm the study led in Section II-F, since u and v are, as expected, lower than $u^* = v^* = 0.142$ ($u \approx 0.087$, $v \approx 0.089$).

In addition, as seen in Section II-E, we have access to an approximation of z^*. Our algorithm yielded a value of

[1]Only results are given for γ and β since α is not servoed.

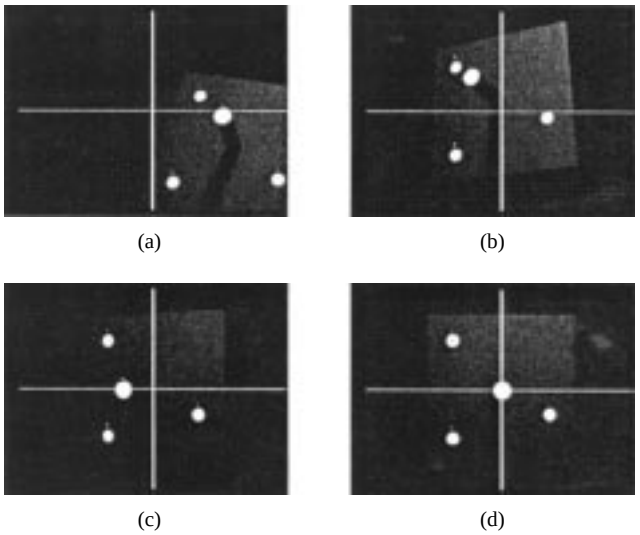

Fig. 8 First experiment. (a) Initial image. (b) Image after the preliminary task. (c) Image after maximization of l. (d) Final image.

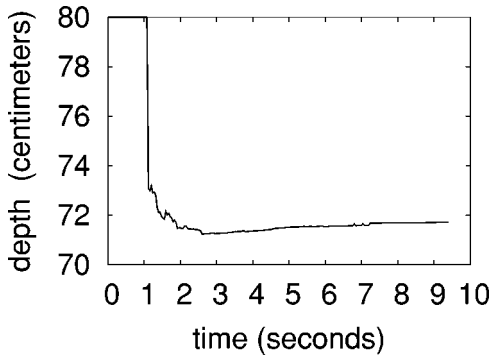

Fig. 9. First experiment: Behavior of $\widehat{z^*}$ during the motion.

71.72 cm, so this method is accurate, since the right value is within [72.70 cm, 72.91 cm] (see the final camera pose). Fig. 9 shows the behavior of $\widehat{z^*}$ during the motion. As soon as a sufficient number of image acquisitions is available, a good approximation is provided. The consequence of a good value for $\widehat{z^*}$ in the control law can be seen in Fig. 7(a) since the error decreases suddenly (near 1.5 s).

Furthermore, once the task is achieved, the camera is in front of the object and $\widehat{z^*}$ is known. Thus, according to perspective projection (1), 3-D features can be reconstructed. For example, for l and h these values have been obtained, $\widehat{l} = 12.92$ cm and $\widehat{h} = 12.89$ cm, instead of $l = h = 13$ cm. Thus, both the algorithms of positioning and reconstruction provide good results.

Twenty other positioning tasks have been realized, such that $\gamma \in [-4°, 22°]$, $\beta \in [8°, 33°]$, $\alpha \in [-25°, 36°]$, and $z_3 \in [50\text{ cm}, 98\text{ cm}]$. Very good results have also been obtained concerning the positioning task, as well as the reconstruction of 3-D features. They are summarized in Table I, in which function $Q(x)$ yields the relative error of the measurement of x in percentage.

The second experiment consists in positioning the camera for a particular desired orientation different from parallel (fol-

TABLE I
RESULTS ON 20 EXPERIMENTS

x	$\overline{x}$	$\max x_i$	$\min x_i$	σ_x
γ	-0.009	0.439	-0.224	0.017
β	-0.489	-0.156	-0.751	0.133
$Q(l)$	0.385	1.078	-1.103	0.493
$Q(h)$	0.365	1.104	-1.102	0.491
$Q(z)$	-0.366	1.109	-1.077	0.489

lowing on from the remark made in Section III-A). In this case, we have the following for Euler's angles:

1) initial camera orientation: $\gamma = 9.08°$, $\beta = 16.45°$;
2) desired camera orientation: $\gamma = -15°$, $\beta = -20°$;
3) final camera orientation: $\gamma = -15.27°$, $\beta = -19.64°$.

Again, these results are satisfactory, since $u = v \approx 0.09$ are lower than u^* and v^*. Figs. 10 and 11 depict the behavior of the control law during the maximization of l and h, respectively, and Fig. 12 describes the different phases to achieve the task. In particular, Figs. 10(d) and 11(d) validate the modeling of l and h. Nevertheless, we remark in Fig. 11(b) a term ω_z introduced by the main task to ensure $\underline{s} = \underline{s}^*$ which did not exist in the first experiment (remember that the interaction matrix given in (38) is computed for a desired position parallel to the object). This term leads a nonplanar motion and, consequently, to a worse approximation for z^*, but accurate enough to achieve the task [here again, $||\underline{s} - \underline{s}^*||$ decreases suddenly when $\widehat{z^*}$ is used in (41)]. Furthermore, the condition $\tilde{P}_k \leq \sigma^2 \nu$ is verified for $t_0 = 0.88$ s for l, and $t_0 = 0.72$ s for h, with a relative error on $\Theta_{xc}(\Theta_{yc})$ of 5.18% (31.52%). Here again, the duration of the maximization of l and h is, more and less, respected: 9.20 and 11.00 s.

Now we consider the case of a coarse-calibrated camera. Errors have been introduced in the intrinsic parameters of the camera. We have added an error of 10% on these parameters, and we have neglected the radial distortion of the lens. The experiment, consisting of achieving a positioning task to a position parallel the object, has led to $\gamma = 0.5°$ and $\beta = -0.86°$ instead of $\gamma = 0.12°$ and $\beta = -0.49°$ for a calibrated camera with the same initial pose. This experiment shows that nonsignificant differences exist concerning the positioning error between a calibrated camera and a coarse-calibrated one. This validates the proof given in Section II-G. As anticipated, our algorithm is sensitive to the eye-to-hand calibration, since errors in the transformation matrix between the end-effector and the camera (5° for rotations, 3 cm for translations) have led to $\gamma = -1.46°$ and $\beta = 1.71°$. However, these results remain satisfactory (since the final orientation error is really inferior to the introduced errors).

Now that we have validated our approach, we will illustrate it on objects with complex shapes. Two examples are given.

IV. APPLICATION TO OBJECTS OF COMPLEX SHAPE

A. Example A

In the proposed example, we use the case of an unknown binary object of complex shape. Its projection yields a contour

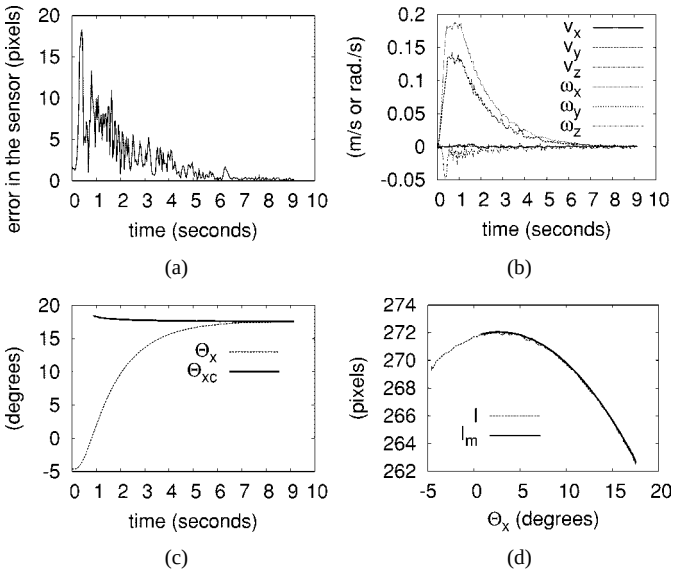

Fig. 10. Second experiment. Servoing of Θ_x.

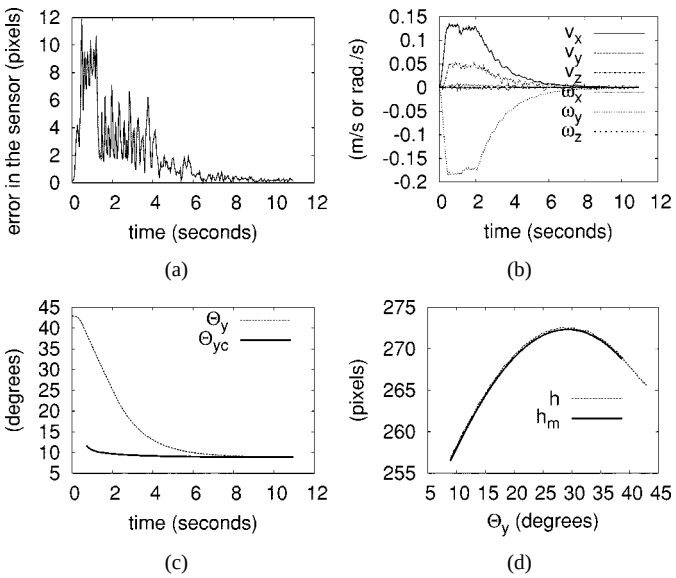

Fig. 11. Second experiment. Servoing of Θ_y.

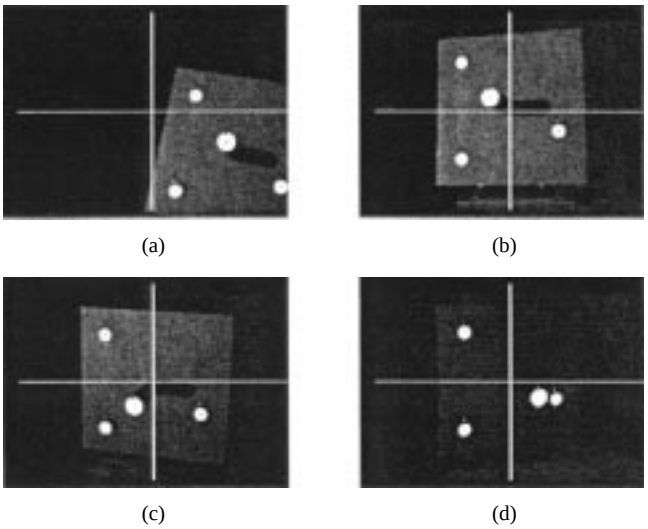

Fig. 12. Second experiment. The different phases to achieve the task.

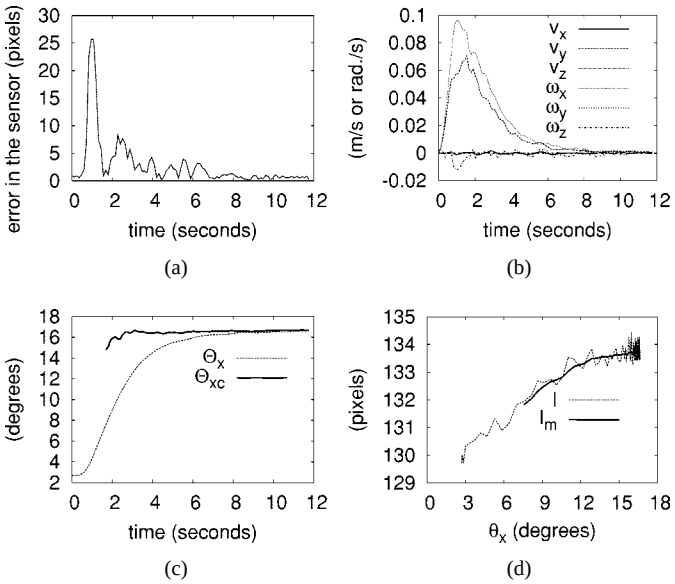

Fig. 13. Example A. Maximization of l.

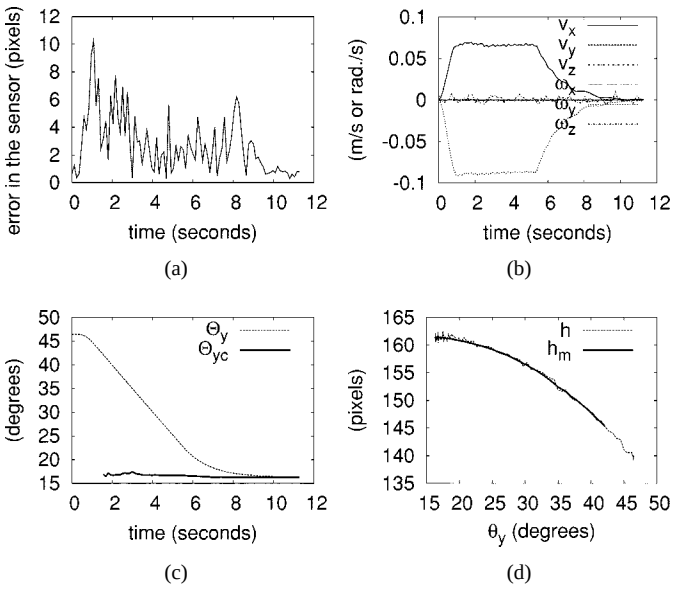

Fig. 14. Example A. Maximization of h.

with eight points of inflexion, from which three are selected, tracked, and used in the control scheme.[2] Our extraction and tracking algorithm is executed in a time of 120 ms. For this reason, we have set $w_{sat} = 5°/s$, $\lambda_{po} = 3$, and $\nu = 250$.

Figs. 13 and 14 describe the behavior of the control law during the maximization of l and h. Fig. 15 summarizes the different phases to achieve the task. The lines linking the three points of inflexion point out the similarity to the previous experimental results (see Fig. 8). Fig. 16 shows the behavior of $\widehat{z^*}$ during the motion. For this experiment we have the following:

1) initial camera orientation: $\gamma = 7.77°$, $\beta = 29.23°$;
2) final camera orientation: $\gamma = -0.90°$, $\beta = 0.24°$.

[2]We verified that points of inflexion in the object are also points of inflexion in the image. In fact, it is shown in [33] that this property is satisfied in the non degenerate cases (which occur when the image of the object becomes a segment).

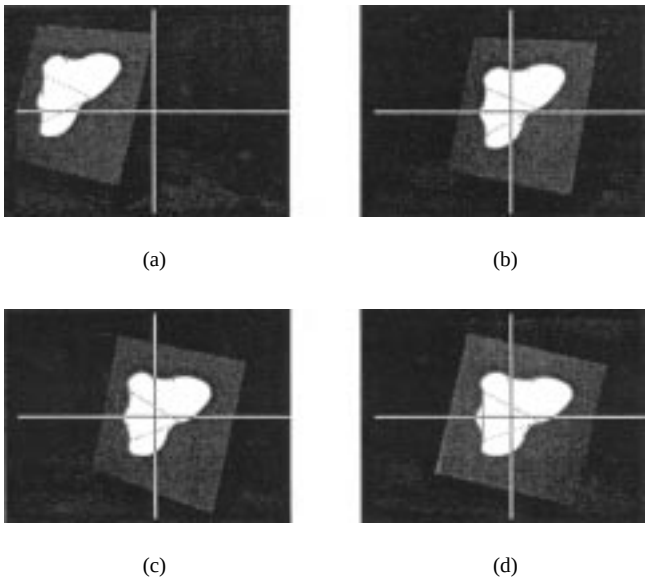

Fig. 15. Example A. The different phases to achieve the task.

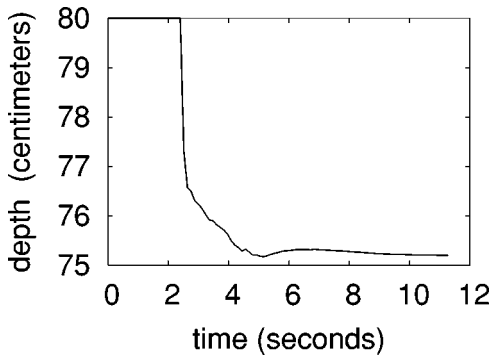

Fig. 16. Example A. Behavior of $\widehat{z^*}$ during the motion.

Therefore, we obtain, in the case of a complex object, good results ($u \approx 0.05$, $v \approx 0.04$), even if during the maximization of l its variation is not very high [see Fig. 13(d)]. Again, in Figs. 13(c) and 14(c), we remark that we quickly obtain a stable desired angular value as well as in Fig. 16 for $\widehat{z^*}$. Furthermore, the condition $\tilde{P}_k \leq \sigma^2\nu$ is verified for $t_0 = 1.68$ s for l, and $t_0 = 1.56$ s for h with a relative error on $\Theta_{xc}(\Theta_{yc})$ of 11.28% (3.82%). On the other hand, as expected, the duration of the maximization of $l(h)$ is around 10 s, more precisely, 11.88 s (11.40 s).

B. Example B

In this last example, we consider the case of a more complex object: a raw ham. This object is moving on a conveyor, and the goal is to place the camera parallel to it to perform an inspection of the cut [37]. To do this, we used the method described in [38] to track at the frame rate (40 ms) the three feature points we need. It is based on the minimization of the sum of squared differences of images intensities between the current window (around the point to track) and another window acquired at the previous frame.

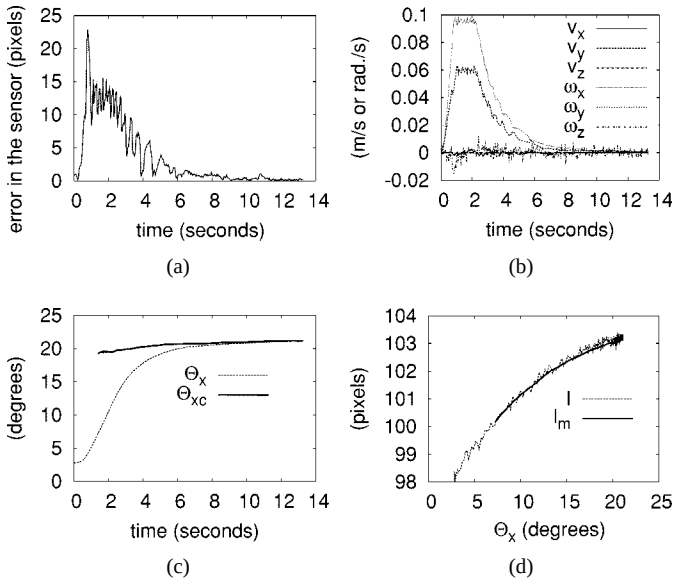

Fig. 17. Example B. Maximization of l.

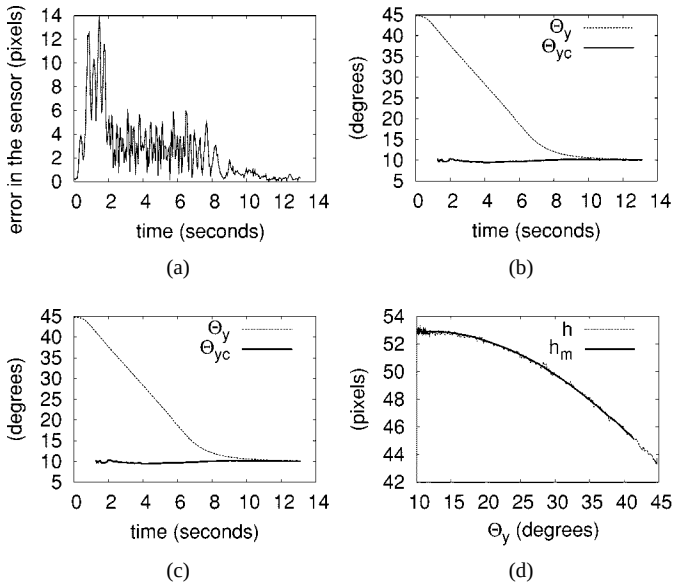

Fig. 18. Example B. Maximization of h.

When more than three feature points exist, we have to select them from the first image only. First, with respect to geometrical conditions, they must be far away from each other, and second, the tracking process must track them during the entire duration of the servoing. For this purpose, we have proposed in [39] an approach based on an entropy criterion to select the points for a robust tracking.

The experimental results are depicted in Figs. 17–20. For this example, since the tracking algorithm necessitates small displacements from one image to another [38], w_{sat}, λ_{pr}, and λ_{po} have been set, respectively, to 5°/s, 0.5, and 5. Let us point out that, again, the online estimations of Θ_{xc} and Θ_{yc} are accurate: 1.37% and 9.28%. However, in this case, it is difficult to measure the positioning accuracy.

Again, we can clearly see in Fig. 18(a) the benefit of using $\widehat{z^*}$ in the control law. In addition, for the application, 3-D features can be obtained to characterize the cut of the ham.

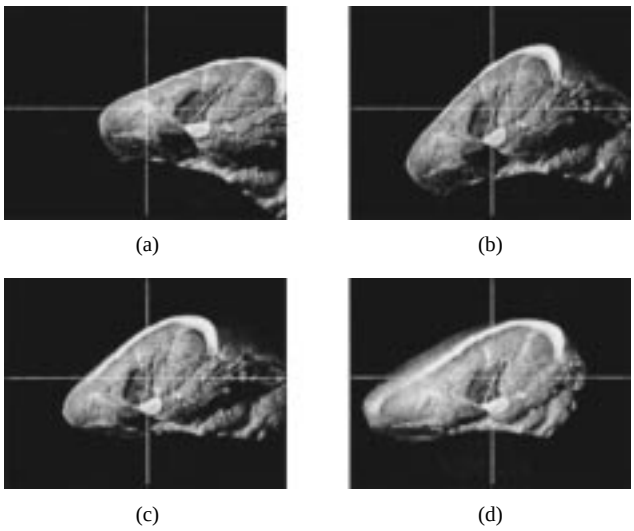

Fig. 19. Example B. The different phases to achieve the task.

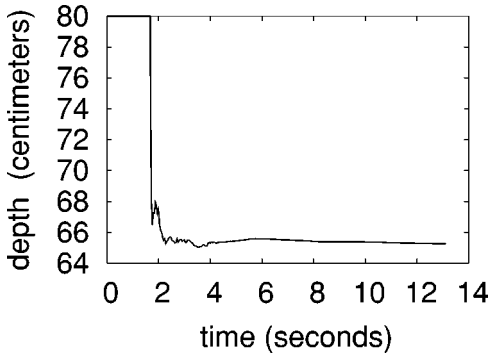

Fig. 20. Example B. Behavior of $\widehat{z^*}$ during the motion.

V. Conclusion

We have presented a way to achieve positioning tasks by 2-D visual servoing when the desired image of the object cannot be precisely described, and for any desired orientation of the camera. However, we have to assume the object to be planar and motionless.

Although we use approximations and an online estimation, we have shown how to ensure the convergence of the control law, consequently leading to accurate positionings ($\approx 0.5°$). In addition, an algorithm based on the modeling of the trajectory of the camera provides the depth between the camera and the object. Hence, in the case of a positioning task consisting of moving the camera parallel to the object, precise 3-D reconstruction can be expected ($\approx$ 0.4%). Moreover, experimental results have shown that our approach is not sensitive to the calibration of the camera. Thus, our method combines in part the advantages of both 2-D visual servoing and 3-D reconstruction methods, in the sense that 3-D information can be obtained without a calibrated camera.

On the other hand, two experimental examples have been given to show the effectiveness of our approach. The first, based on the extraction of inflexion points, has shown that an accurate positioning can be obtained at a relative high rate (120 ms), and the second, based on points of interest, on a more complex object at the frame rate (40 ms).

Finally, we think that this paper contributes to an expansion of the application area of visual servoings, in the sense that complex objects or objects of unknown shape can now be treated even if the desired image is not precisely known.

ACKNOWLEDGMENT

The authors wish to thank A. Compert for his contribution to the improvement of the readability of this paper.

REFERENCES

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 651–670, Oct. 1996.
[2] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 313–326, June 1992.
[3] W. Jang and Z. Bien, "Feature-based visual servoing of an eye-in-hand robot with improved tracking performance," in *Proc. IEEE Int. Conf. Robotics and Automation, ICRA'91*, Sacramento, CA, Apr. 1991, pp. 2254–2260.
[4] N. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Vision and control techniques for robotic visual tracking," in *Proc. IEEE Int. Conf. Robotics and Automation, ICRA'91*, Sacramento, CA, Apr. 1991, pp. 857–864.
[5] K. Hashimoto, T. Ebine, and H. Kimura, "Visual servoing with hand-eye manipulator—Optimal control approach," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 766–774, Oct. 1996.
[6] P. I. Corke and S. Hutchinson, "A new hybrid image-based visual servo control scheme," in *Proc. 39th IEEE Int. Conf. Decision and Control, CDC 2000*, Sydney, Australia, Dec. 2000, pp. 2521–2526.
[7] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 684–696, Oct. 1996.
[8] N. Daucher, M. Dhome, J. T. Lapresté, and G. Rives, "Speed command of a robotic system by monocular pose estimate," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, IROS'97*, Grenoble, France, Sept. 1997, pp. 55–62.
[9] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images, a review," *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.
[10] G. Adiv, "Inherent ambiguities in recovering 3d motion and structure from a noisy flow field," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 477–489, May 1989.
[11] J. L. Crowley, P. Stelmaszyk, and P. Puget, "Measurement and integration of 3d structures by tracking edges lines," *Int. J. Comput. Vis.*, vol. 8, pp. 29–52, July 1992.
[12] Y. Aloimonos, I. Weiss, and A. Bandopadhay, "Active vision," *Int. J. Comput. Vis.*, vol. 1, pp. 333–356, Jan. 1987.
[13] A. Bandopadhay, B. Chandra, and D. H. Ballard, "Egomotion using active vision," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, CVPR'86*, June 1986, pp. 498–503.
[14] B. Espiau and P. Rives, "Closed-loop recursive estimation of 3d features for a mobile vision system," in *Proc. IEEE Int. Conf. Robotics and Automation, ICRA'87*, Raleigh, NC, Apr. 1987, pp. 1436–1443.
[15] R. Bajcsy, "Active perception," *Proc. IEEE*, vol. 76, pp. 996–1005, Aug. 1988.
[16] G. Sandini and M. Tistarelli, "Active tracking strategy for monocular depth inference over multiple frames," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 13–27, Jan. 1990.
[17] K. Kutulakos and C. Dyer, "Recovering shape by purposive viewpoint adjustment," *Int. J. Comput. Vis.*, vol. 12, pp. 113–136, Feb. 1994.
[18] R. Cipolla, *Active Visual Inference of Surface Shape*. New York: Springer-Verlag, 1995, vol. 1016, LNCS.
[19] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, "Structure from controlled motion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 492–504, May 1996.
[20] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 404–417, Oct. 1987.
[21] H. Hashimoto, T. T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proc. IEEE Int. Conf. Robotics and Automation, ICRA'91*, Sacremento, CA, Apr. 1991, pp. 2267–2272.
[22] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *Third Int. Symp. Experimental Robotics, ISER'93*, Kyoto, Japan, Oct. 1993, pp. 187–193.
[23] J. L. Crowley, M. Mesrabi, and F. Chaumette, "Comparaison of kinematic and visual servoing for fixation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, IROS'95*, vol. 1, Pittsburgh, PA, Aug. 1995, pp. 335–341.
[24] G. D. Hager, W. C. Chang, and A. S. Morse, "Robot hand-eye coordination based on stereo vision," *IEEE Control Syst. Mag.*, vol. 15, pp. 30–39, Feb. 1995.
[25] R. Kelly, "Robust asymptotically stable visual servoing of planar robots," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 759–766, Oct. 1996.
[26] N. P. Papanikolopoulos, B. Nelson, and P. K. Khosla, "Six degree-of-freedom hand/eye visual tracking with uncertain parameters," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 725–732, Oct. 1995.
[27] B. Bishop and M. W. Spong, "Adaptive calibration and control of 2d monocular visual servo systems," in *Proc. 5th IFAC Symp. Robot Control, SYROCO'97*, vol. 2, Nantes, France, Sept. 1997, pp. 525–530.
[28] E. Malis, F. Chaumette, and S. Boudet, "2 1/2d visual servoing," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 238–250, Apr. 1999.
[29] B. Yoshimi and P. K. Allen, "Active uncalibrated visual servoing," in *Proc. IEEE Int. Conf. Robotics and Automation, ICRA'94*, San Diego, CA, May 1994, pp. 156–161.
[30] A. Crétual and F. Chaumette, "Positioning a camera parallel to a plane using dynamic visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, IROS'97*, Grenoble, France, Sept. 1997, pp. 43–48.
[31] J. M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *J. Vis. Commun. Image Represent.*, vol. 6, pp. 348–365, Dec. 1995.
[32] C. Samson, B. Espiau, and M. Le Borgne, *Robot Control: The Task Function Approach*. Oxford, U.K.: Clarendon, 1991.
[33] C. Collewet, "Contributions à l'élargissement du champ applicatif des asservissements visuels 2D," Ph.D. dissertation, Univ. of Rennes I, Rennes, France, 1999.
[34] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[35] P. Y. Oh and P. K. Allen, "Visual servoing by partioning degrees of freedom," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 1–17, Feb. 2001.
[36] D. F. DeMenthon and L. S. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comput. Vis.*, vol. 15, pp. 123–141, 1995.
[37] D. Legeard, P. Marty-Mahé, J. Camillerapp, P. Marchal, and C. Leredde, "Real-time quality evaluation of pork hams by color machine vision," in *Proc. SPIE Machine Vision Application in Industrial Inspection VII, Electronic Imaging'99*, vol. 3652, San Jose, CA, Jan. 1999.
[38] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, CVPR'94*, Seattle, WA, June 1994, pp. 593–600.
[39] C. Kermad and C. Collewet, "Improving feature tracking by robust points of interest selection," in *6th Int. Fall Workshop on Vision, Modeling and Visualization, VMV'2001*, Stuttgart, Germany, Nov. 2001, pp. 415–423.

Christophe Collewet was born in Bray-Dunes, France, in 1962. He graduated in automatic control from the École Nationale Supérieure d'Ingénieurs de Caen, Caen, France, in 1986 and received the Ph.D. degree in signal processing from the University of Rennes, Rennes, France, in 1999.

Since 1988, he has been with Cemagref, Rennes, France, the French Institute of Agricultural and Environmental Engineering Research, as "Chargé de Recherche." His research interests include robotics, image processing, and vision-based control.

François Chaumette was born in Nantes, France, in 1963. He graduated from the École Nationale Supérieure de Mécanique, Nantes, France, in 1987 and received the Ph.D. degree and "Habilitation à Diriger des Recherches" in computer science from the University of Rennes, Rennes, France, in 1990 and 1998, respectively.

Since 1990, he has been with IRISA/INRIA, Rennes, France, where he is now "Directeur de Recherches." His research interests include robotics, computer vision, and, in particular, the coupling of these two research domains (visual servoing and active perception).

Dr. Chaumette received the AFCET/CNRS Prize for the best French thesis in automatic control in 1991. He is currently an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION.