Point-Based and Region-Based Image Moments for Visual Servoing of Planar Objects

Omar Tahri and François Chaumette, *Member, IEEE*

***Abstract*—Moments are generic (and usually intuitive) descriptors that can be computed from several kinds of objects, defined either from closed contours or from a set of points. In this paper, we present improvements in image-based visual servo using image moments. First, the analytical form of the interaction matrix related to the moments computed from a set of coplanar points is derived, and we show that it is different from the form obtained previously, using coplanar closed contours. Six visual features are selected to design a decoupled control scheme when the object is parallel to the image plane. This nice property is then generalized to the case where the desired object position is not parallel to the image plane. Finally, experimental results are presented to illustrate the validity of our approach and its robustness, with respect to modeling errors.**

***Index Terms*—Image moment, invariant, visual servoing.**

I. Introduction

TO DATE, an open question in image-based visual servoing is to determine the set of visual features to be used in the control scheme so that an optimal behavior of the system is obtained. In the past, most works were concerned with simple objects, and the features used as input of the control scheme were generally coordinates of points, or parameters describing the configuration in the image of segments, straight lines, or ellipses [9], [13]. The set of objects to which these methods can be applied is, thus, limited. They have also the basic requirement of feature matching between initial and desired images. More recent works have tried to surmount the problems mentioned above, by using, for example, an eigenspace representation [8], the polar signature of an object contour [5], or, as in this paper, image moments [4]. Image moments are particularly interesting. They can be computed easily from a binary or a segmented image, or from a set of extracted points of interest. Computing moments in several images (for visual servoing, the initial and the desired ones, and all the successive images acquired by the camera) just necessitates a high-level global matching of the object, and not an accurate and tedious matching of each object point. They are generic, describing the same geometrical entities, whatever the object shape complexity. Low-order moments have an intuitive meaning, since they are directly related to the area, the centroid, the inertial moments, and the orientation of the object in the image. Image moments have been widely studied in the computer vision community [12], [21] [22], especially for pattern recognition applications. In addition to the advantages recalled above, judicious combinations of moments are indeed invariant to some transformations, such as scale, 2-D translation, and/or 2-D rotation. This property is of great interest in pattern recognition, which explains the amount of work on the determination of moments invariants (see [2], [19], [23], and [32], for instance). As it will be shown in this paper, such an invariance property is also of particular interest in visual servoing.

Whatever the nature of the possible measures extracted from the image, from a set of image points coordinates to a set of image moments, the main question is how to combine them to obtain an adequate behavior of the system. In most works, the combination is nothing but a simple stacking. If the error between the initial value of the features and the desired one is small, and if the task to realize constrains all the available degrees of freedom (DOFs), that may be a good choice. However, as soon as the error is large, problems may appear, such as reaching local minimum or a task singularity [3]. To overcome these problems, combining path planning and visual servoing is a first approach, since tracking planned trajectories allows the error to always remain small [7], [20], [34]. A second approach is to use the measures to build particular visual features that will ensure expected properties to the control scheme. Position-based visual servoing belongs to this approach. Using visual measures and an *a priori* knowledge of a computer-aided design (CAD) model of the observed object, the relative camera-object pose is estimated and used as input to the control scheme [33]. An adequate 3-D trajectory can thus be obtained, such as a geodesic for the orientation and a straight line for the translation. However, position-based visual servoing may suffer from potential instabilities due to image noise [3] (the "cooking" of the visual features from the measures is quite complex, since it comes from the solution of an inverse problem).[1] 2–1/2-D (or hybrid) visual servoing also belongs to this approach [18]. The goals were to combine 2-D features and 3-D features to decouple the control of the rotation from the control of the translation (as in 3-D visual servoing), to ensure as much as possible the visibility of the object in the camera field of view (FOV), and also to demonstrate the global stability of the system using only the measures from the current and the desired images. Finally, several works have been realized in image-based visual servoing

Manuscript received July 30, 2004; revised January 13, 2005. This paper was recommended for publication by Associate Editor J. Kosecka and Editor F. Park upon evaluation of the reviewers' comments. This paper was presented in part at the International Conference on Robotics and Automation, Taipei, Taiwan, R.O.C., September 2003, and in part at the International Conference on Robotics and Automation, New Orleans, LA, April 2004.

O. Tahri was with IRISA/INRIA Rennes, 35 042 Rennes-Cedex, France. He is now with CEA/LIST, 92265 Fontenay-aux-Roses, France.

F. Chaumette is with IRISA/INRIA Rennes, 35 042 Rennes-Cedex, France (e-mail: Francois.Chaumette@irisa.fr).

Digital Object Identifier 10.1109/TRO.2005.853500

[1]The marvelous expression "cooking of features" is from Peter Corke.

following the same general objective. In [14], the coordinates of points are expressed in a cylindrical coordinate system, instead of the classical Cartesian one, to improve the robot trajectory. In [11], the three coordinates of the centroid of an object in a virtual image obtained through a spherical projection have been selected to control three DOFs of an underactuated system. This selection ensures a passivity property that is useful in controlling such complex systems. In [24], a vanishing point and the horizon line have been selected to control a similar system. This choice ensures a good decoupling between translational and rotational DOFs. In [15], vanishing points have also been used for a dedicated object (a 3-D rectangle), once again for decoupling properties. For the same object, six visual features have been designed in [6] to control the six DOFs of a robot arm, following a partitioned approach. Finally, in [4], a first attempt to select six features from image moments has been recently presented. This selection was valid only for planar objects whose desired pose is parallel to the image plane. In this paper, the selection method is significantly improved, and is generalized for any desired object pose.

In most related previous works, the selection of the visual features was guided by a partitioned approach to design a decoupled control scheme, that is, to try to associate each DOF to be controlled with only one visual feature. That is indeed a good strategy. However, it is not sufficient to obtain a good behavior of the visual features and of the robot trajectory simultaneously. In few words, we recall that the time variation $\dot{\mathbf{s}}$ of the visual features $\mathbf{s}$ can be expressed linearly with respect to the relative camera-object kinematics screw $\mathbf{v}$

$$\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v} \tag{1}$$

where $\mathbf{L_s}$ is the interaction matrix related to $\mathbf{s}$ [9], [13]. The control scheme is usually designed to try to ensure an exponential decoupled decrease of the visual features to their desired value $\mathbf{s}^*$, from which we deduce, if we consider an eye-in-hand system observing a static object

$$\mathbf{v_c} = -\lambda \widehat{\mathbf{L_s}}^{+}(\mathbf{s} - \mathbf{s}^*) \tag{2}$$

where $\widehat{\mathbf{L_s}}$ is a model or an approximation of $\mathbf{L_s}$, $\widehat{\mathbf{L_s}}^{+}$ is the pseudoinverse of $\widehat{\mathbf{L_s}}$, λ is a positive gain tuning the time to convergence, and $\mathbf{v_c}$ is the camera velocity sent to the low-level robot controller. In the following, we denote $\boldsymbol{\upsilon}$ and $\boldsymbol{\omega}$, respectively, as the translational and rotational components of the kinematic screw, so that $\mathbf{v_c} = (\boldsymbol{\upsilon}, \boldsymbol{\omega}) = (\upsilon_x, \upsilon_y, \upsilon_z, \omega_x, \omega_y, \omega_z)$. An exponential decoupled decrease will be obtained simultaneously on the visual features and on the camera velocity (that would give a perfect behavior) only if $\mathbf{L_s}$ and $\widehat{\mathbf{L_s}}^{+}$ are constant. If it is possible to choose $\widehat{\mathbf{L_s}}^{+}$ as a constant matrix, the form of $\mathbf{L_s}$ is set by the design of $\mathbf{s}$, and it is generally very far from a constant matrix. For instance, the well-known interaction matrix related to the coordinates $\mathbf{x} = (x, y)$ of an image point is given by

$$\mathbf{L_x} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -1-x^2 & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix} \tag{3}$$

where Z is the depth of the observed point. We can see from (3) that the velocities $\dot{x}$ and $\dot{y}$ are really not the same with respect to each camera velocity component. Some are inversely proportional to the depth Z of the point, some are linearly dependent on the image coordinates, while others depend on them at the second order. The nonlinearities in system (1) using (3) explain the difference of behaviors in image space and in 3-D space, and the inadequate robot trajectory that occurs sometimes when the displacement to realize is large [3] (of course, for small displacements, such that the variations of $\mathbf{L_x}$ are negligible, a correct behavior is obtained, as already noted above). An important problem is, thus, to determine visual features, such that their interaction matrix minimizes the nonlinearities in (1). Ideally, we would like $\mathbf{L_s}$ to be the identity matrix $\mathbf{I}_6$, even if this goal is probably impossible to reach. Note that designing a decoupled or a partitioned system is a step toward this goal, since it introduces terms equal to zero in $\mathbf{L_s}$.

In this paper, we propose significant improvements in the determination of adequate visual features using image moments. In the next section, we first briefly recall the basic definition of image moments. We then give a general analytical form of their interaction matrix. In [4], only objects defined from closed contours were considered, while this paper deals also with moments computed from a set of image points. Section III is devoted to the determination of six visual features to control the six DOFs of the system for the particular case where the desired object pose is parallel to the image plane. This result is generalized in Section IV to the case where the object may have any orientation with respect to the camera. Finally, experimental results are presented in Section V to validate the proposed theoretical results.

II. Modeling

A. Moment Invariants

If we consider a dense object $\mathcal{O}$ in the image, defined by a set of closed contours [see, for instance, Figs. 2(b) and 8(b)], its 2-D moments m_{ij} of order $i + j$ are defined by

$$m_{ij} = \iint_{\mathcal{O}} x^i y^j dx dy. \tag{4}$$

The centered moments μ_{ij} are computed with respect to the object entroid (x_g, y_g). They are defined by

$$\mu_{ij} = \iint_{\mathcal{O}} (x - x_g)^i (y - y_g)^j dx dy \tag{5}$$

where $x_g = m_{10}/a$ and $y_g = m_{01}/a$, $a = m_{00}$ being the object area. Similarly, for a discrete set of n image points [see, for instance, Fig. 14(a)], the moments are defined by

$$m_{ij} = \sum_{k=1}^{n} x_k^j y_k^j \tag{6}$$

while the centered moments are now given by

$$\mu_{ij} = \sum_{k=1}^{n} (x_k - x_g)^i (y_k - y_g)^j \tag{7}$$

where $x_g = m_{10}/n$ and $y_g = m_{01}/n$ ($m_{00} = n$ in that case). The centered moments defined either from (5) or (7) are known to be invariant to 2-D translational motion. In the literature, many works have presented various methods to derive moment invariants to other transformations, such as scale and 2-D rotation. For instance, moment invariants to rotation have been obtained from radial and angular moments [23], Zernike moments

[2], [30], [32], and complex moments [1], [10]. As for invariants to scale, several combinations of moments have been proposed, such as, for example [19]

$$I_s = \frac{m_{pq}}{m_{00}^{\frac{(p+q+2)}{2}}}. \tag{8}$$

This formula will be used in Section III to decouple the features involved in the control of the translational DOFs v_x and v_y. We now present several combinations of moments that are invariant to 2-D translation, 2-D rotation, and to scale. Complete details on how they have been determined can be found in [21], [22], and [27]

$$\begin{aligned} &c_1 = \frac{I_1}{I_2},\ c_2 = \frac{I_3}{I_4},\ c_3 = \frac{I_5}{I_6},\ c_4 = \frac{I_7}{I_6},\ c_5 = \frac{I_8}{I_6},\\ &c_6 = \frac{I_9}{I_6},\ c_7 = \frac{I_{11}}{I_{10}},\ c_8 = \frac{I_{12}}{I_{10}},\ c_9 = \frac{I_{13}}{I_{15}},\ c_{10} = \frac{I_{14}}{I_{15}} \end{aligned} \tag{9}$$

where invariants I_1 to I_{15} are given in the Appendix. We will see in Sections III and V that depending on the object considered, two among the invariants (9) will be selected as visual features to control the rotational velocities ω_x and ω_y. We now derive a general form of the interaction matrix related to image moments.

B. Interaction Matrix of Image Moments

In [4], the interaction matrix $\mathbf{L}_{m_{ij}}$ related to any moment m_{ij} defined from (4) has been determined. It has been obtained from the following equation:

$$\dot{m}_{ij} = \iint_{\mathcal{O}} \left[\frac{\partial f}{\partial x}\dot{x} + \frac{\partial f}{\partial y}\dot{y} + f(x,y)\left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y}\right)\right] dxdy \tag{10}$$

where $f(x,y) = x^i y^j$. If a planar object is considered, and if we exclude the degenerate case where the camera optical center belongs to this plane, so that, for any object point

$$\frac{1}{Z} = Ax + By + C \tag{11}$$

we obtain (see [4] for more details)

$$\mathbf{L}_{m_{ij}} = [m_{vx}\ m_{vy}\ m_{vz}\ m_{wx}\ m_{wy}\ m_{wz}] \tag{12}$$

where

$$\begin{cases} m_{vx} = -i(Am_{ij} + Bm_{i-1,j+1} + Cm_{i-1,j}) - \delta Am_{ij} \\ m_{vy} = -j(Am_{i+1,j-1} + Bm_{ij} + Cm_{i,j-1}) - \delta Bm_{ij} \\ m_{vz} = (i+j+3\delta)(Am_{i+1,j} + Bm_{i,j+1} + Cm_{ij}) - \delta Cm_{ij} \\ m_{wx} = (i+j+3\delta)m_{i,j+1} + jm_{i,j-1} \\ m_{wy} = -(i+j+3\delta)m_{i+1,j} - im_{i-1,j} \\ m_{wz} = im_{i-1,j+1} - jm_{i+1,j-1} \end{cases} \tag{13}$$

with $\delta = 1$. We now consider the case of moments defined by (6). We will see that a different analytical form of $\mathbf{L}_{m_{ij}}$ is obtained, characterized by the value of δ. Computing the time derivative of (6), we obtain

$$\dot{m}_{ij} = \sum_{k=1}^{n} \left(i x_k^{i-1} y_k^j \dot{x}_k + j x_k^i y_k^{j-1} \dot{y}_k\right). \tag{14}$$

The velocity of any image point $\mathbf{x}_k$ is given from (1) and (3), setting $\mathbf{s} = \mathbf{x}_k$ in (1). More precisely, using (11), the velocity of $\mathbf{x}_k$ can be written

$$\begin{cases} \dot{x_k} = -(Ax_k + By_k + C)v_x \\ \qquad + x_k(Ax_k + By_k + C)v_z \\ \qquad + x_k y_k \omega_x - \left(1 + x_k^2\right)\omega_y + y_k \omega_z \\ \dot{y_k} = -(Ax_k + By_k + C)v_y \\ \qquad + y_k(Ax_k + By_k + C)v_z \\ \qquad + \left(1 + y_k^2\right)\omega_x - x_k y_k \omega_y - x_k \omega_z. \end{cases} \tag{15}$$

Finally, using (15) in (14), and then using (6), we obtain, after simple developments, the interaction matrix related to m_{ij}. Its analytical form is again given by (12) and (13), but with $\delta = 0$.

Matrix $\mathbf{L}_{m_{ij}}$ is, thus, not exactly the same, if we consider the moments (4) of a dense object (i.e., defined by closed contours), or the moments (6) of a discrete object (i.e., defined by a set of discrete points). The analytical forms are similar, since the two terms of (14) correspond exactly to the first two terms present in (10). On the other hand, they are different, since the third term of (10) does not appear in (14). To illustrate these differences on a simple example, let us consider moment m_{00}. In the discrete case, m_{00} is nothing but the number n of object points. This number is, of course, invariant, and we can check by setting $i = j = \delta = 0$ in (13) that all the terms of $\mathbf{L}_{m_{00}}$ are indeed equal to zero. In the dense case, m_{00} is nothing but the area of the object, and general robot motion modifies its value, as can be checked from (13) using $\delta = 1$.

Many visual features derived from moments have, however, a very similar behavior in both cases. For instance, we can easily compute from (12) the interaction matrix related to the coordinates x_g and y_g of the object center of gravity. We obtain

$$\begin{aligned} \mathbf{L}_{x_g} &= \left[-\frac{1}{Z_g}\ 0\ x_{g_{vz}}\ x_{g_{wx}}\ x_{g_{wy}}\ y_g\right] \\ \mathbf{L}_{y_g} &= \left[0\ -\frac{1}{Z_g}\ y_{g_{vz}}\ y_{g_{wx}}\ y_{g_{wy}}\ -x_g\right] \end{aligned}$$

$$\text{where} \quad \begin{cases} \dfrac{1}{Z_g} = Ax_g + By_g + C \\ x_{g_{vz}} = \dfrac{x_g}{Z_g} + A\epsilon n_{20} + B\epsilon n_{11} \\ y_{g_{vz}} = \dfrac{y_g}{Z_g} + A\epsilon n_{11} + B\epsilon n_{02} \\ x_{g_{wx}} = -y_{g_{wy}} = x_g y_g + \epsilon n_{11} \\ x_{g_{wy}} = -\left(1 + x_g^2 + \epsilon n_{20}\right) \\ y_{g_{wx}} = 1 + y_g^2 + \epsilon n_{02} \end{cases} \tag{16}$$

with $n_{ij} = \mu_{ij}/m_{00}$, $\epsilon = 4$ for dense objects, and $\epsilon = 1$ for discrete objects.

Similarly, if we consider the centered moments defined by (5) or (7), we obtain, after tedious developments

$$\mathbf{L}_{\mu_{ij}} = [\mu_{vx}\ \mu_{vy}\ \mu_{vz}\ \mu_{wx}\ \mu_{wy}\ \mu_{wz}] \tag{17}$$

with

$$\begin{cases} \mu_{vx} = -(i+\delta)A\mu_{ij} - iB\mu_{i-1,j+1} \\ \mu_{vy} = -jA\mu_{i+1,j-1} - (j+\delta)B\mu_{ij} \\ \mu_{vz} = -A\mu_{wy} + B\mu_{wx} + (i+j+2\delta)C\mu_{ij} \\ \mu_{wx} = (i+j+3\delta)\mu_{i,j+1} + (i+2j+3\delta)y_g\mu_{ij} \\ \qquad + ix_g\mu_{i-1,j+1} - i\epsilon n_{11}\mu_{i-1,j} - j\epsilon n_{02}\mu_{i,j-1} \\ \mu_{wy} = -(i+j+3\delta)\mu_{i+1,j} - (2i+j+3\delta)x_g\mu_{ij} \\ \qquad - jy_g\mu_{i+1,j-1} + i\epsilon n_{20}\mu_{i-1,j} + j\epsilon n_{11}\mu_{i,j-1} \\ \mu_{wz} = i\mu_{i-1,j+1} - j\mu_{i+1,j-1}. \end{cases}$$

In all cases, and as expected, we can check from μ_{vx} and μ_{vy} that all centered moments are invariant, with respect to translational motions parallel to the image plane when the object is parallel to the image plane ($\mu_{vx} = \mu_{vy} = 0$ if $A = B = 0$). Similarly, for the same configurations, we can check from (12) that the invariants to scale given by (8) are invariant to translational motion along the optical axis ($I_{s_{v_z}} = 0$ if $A = B = 0$). Finally, after quite tedious computations, we can also check that the invariants c_i given in (9) are such that $c_{i_{\omega_z}} = 0$, and, if $A = B = 0$, such that $c_{i_{v_x}} = c_{i_{v_y}} = c_{i_{v_z}} = 0$. As detailed in the next section, these invariance properties will be useful in selecting adequate visual features for visual servoing.

III. Choice of the Visual Features

In this section, we select from the previous theoretical results six combinations of moments to control the six DOFs of the robot. Our objective is to obtain a sparse interaction matrix that changes slowly around the desired position of the camera. We will see that the solution we present is such that the interaction matrix is block-triangular when the object is parallel to the image plane. Furthermore, we will see that for the same positions, the elements corresponding to translational motions form a constant diagonal block, which is independent of depth. In [4], this last interesting property was not satisfied.

We first assume that the desired position of the object is parallel to the image plane (i.e., $A = B = 0$) and we denote $\mathbf{L}_{\mathbf{s}}^{\parallel}$ the interaction matrix for such configurations. The more general case where the desired object position may have any orientation with respect to the image plane will be treated in the next section.

A. Visual Features to Control the Translational DOFs

In [4] and [6], the three visual features used to control the translational DOFs have been selected as the coordinates x_g, y_g of the center of gravity, and the area $a = m_{00}$ of the object in the image. In that case, we obtain from (16) and (12)

$$\begin{aligned} \mathbf{L}_{x_g}^{\parallel} &= [-C \;\; 0 \;\; Cx_g \;\; \epsilon_1 \;\; -(1+\epsilon_2) \;\; y_g] \\ \mathbf{L}_{y_g}^{\parallel} &= [0 \;\; -C \;\; Cy_g \;\; 1+\epsilon_3 \;\; -\epsilon_1 \;\; -x_g] \\ \mathbf{L}_{a}^{\parallel} &= [0 \;\; 0 \;\; 2a\delta C \;\; 3a\delta y_g \;\; -3a\delta x_g \;\; 0] \end{aligned} \tag{18}$$

with $\epsilon_1 = x_g y_g + \epsilon n_{11}$, $\epsilon_2 = x_g^2 + \epsilon n_{20}$, and $\epsilon_3 = y_g^2 + \epsilon n_{02}$. First, we recall that if a set of n points is considered, we cannot use the area m_{00}, since it is a constant value equal to n. This case will be studied later. Then, we can note that, even if the above matrix is triangular, most of its elements are not constant. Moreover, the third feature a does not have the same dynamics with respect to v_z as x_g and y_g with respect to v_x and v_y, respectively.

A better choice can be obtained from these intuitive features by just adding an adequate normalization. More precisely, we define

$$a_n = Z^* \sqrt{\frac{a^*}{a}}, \quad x_n = a_n x_g, \quad y_n = a_n y_g \tag{19}$$

where a^* is the desired area of the object in the image, and Z^* the desired depth between the camera and the object. The interaction matrices related to these normalized features can be easily determined from (18). Noting that $Z^*\sqrt{a^*} = Z\sqrt{a} = \sqrt{S}$, where S is the area of the planar object, we obtain

$$\begin{aligned} \mathbf{L}_{x_n}^{\parallel} &= [-1 \;\; 0 \;\; 0 \;\; a_n\epsilon_{11} \;\; -a_n(1+\epsilon_{12}) \;\; y_n] \\ \mathbf{L}_{y_n}^{\parallel} &= [0 \;\; -1 \;\; 0 \;\; a_n(1+\epsilon_{21}) - a_n\epsilon_{22} \;\; -x_n] \\ \mathbf{L}_{a_n}^{\parallel} &= [0 \;\; 0 \;\; -1 \;\; -a_n\epsilon_{31} \;\; a_n\epsilon_{32} \;\; 0] \end{aligned} \tag{20}$$

with $\epsilon_{11} = \epsilon_{22} = 4n_{11} - x_g y_g/2$, $\epsilon_{12} = 4n_{20} - x_g^2/2$, $\epsilon_{21} = 4n_{02} - y_g^2/2$, $\epsilon_{31} = 3y_g/2$, and $\epsilon_{32} = 3x_g/2$. Since a_n is inversely proportional to $\sqrt{a}$, we find again the recent result given in [16], stating that the variation of such features depends linearly on the depth (note the constant term in the third element of $\mathbf{L}_{a_n}^{\parallel}$). The normalization by $Z^*\sqrt{a^*}$ has just been chosen, so that this constant term is equal to -1. Furthermore, the design of x_n and y_n allows us to completely partition the three selected features to the three translational DOFs. This decoupling property was expected from (8). We also obtain the same dynamics for the three features [note the diagonal block equal to $-\mathbf{I}_3$ in (20)]. This very nice property will allow us to obtain an adequate robot translational trajectory.

We now consider the case of a discrete object. Since $\mu_{20}+\mu_{02}$ is invariant to 2-D translation and 2-D rotation, we propose to replace in (19) the area and its desired value by

$$a = \mu_{20} + \mu_{02} \text{ and } a^* = \mu_{20}^* + \mu_{02}^*. \tag{21}$$

In that case, the interaction matrix related to x_n, y_n, and a_n is again given by (20), but with

$$\begin{cases} \epsilon_{11} = n_{11} + x_g(y_g - \epsilon_{31}), \; \epsilon_{12} = n_{20} + x_g(x_g - \epsilon_{32}) \\ \epsilon_{21} = n_{02} + y_g(y_g - \epsilon_{31}), \; \epsilon_{22} = n_{11} + y_g(x_g - \epsilon_{32}) \\ \epsilon_{31} = y_g + \dfrac{(y_g\mu_{02} + x_g\mu_{11} + \mu_{21} + \mu_{03})}{a} \\ \epsilon_{32} = x_g + \dfrac{(x_g\mu_{20} + y_g\mu_{11} + \mu_{12} + \mu_{30})}{a}. \end{cases}$$

The control of the translational DOFs can thus be realized with the same nice properties for both cases. We now consider the rotational DOFs.

B. Visual Features to Control the Rotational DOF

First, as in [4] and [6], we use the object orientation α that can be defined from the second-order centered moments: $\alpha = (1/2)\arctan(2\mu_{11}/\mu_{20} - \mu_{02})$. We also use two moment invariants c_i and c_j, chosen in (9). The interaction matrices related to these features have the following form:

$$\begin{aligned} \mathbf{L}_{c_i}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & c_{i_{wx}} & c_{i_{wy}} & 0 \end{bmatrix} \\ \mathbf{L}_{c_j}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & c_{j_{wx}} & c_{j_{wy}} & 0 \end{bmatrix} \\ \mathbf{L}_{\alpha}^{\parallel} &= [0 \;\; 0 \;\; 0 \;\; \alpha_{wx} \;\; \alpha_{wy} \;\; -1] \end{aligned} \tag{22}$$

where the analytical form of $c_{i_{wx}}$, $c_{i_{wy}}$, $c_{j_{wx}}$, and $c_{j_{wy}}$ can be obtained after tedious developments using (17), and where α_{wx} and α_{wy} are shown in the equation at the bottom of the page, with $\beta = 5$ and $\gamma = 1$ for a dense object, and with $\beta = 4$ and $\gamma = 2$ for a discrete object.

Unfortunately, we have not been able to find two combinations of moments c_i and c_j such that $c_{i_{wy}} = c_{j_{wx}} = 0$, and such that $c_{i_{wx}}$ and $c_{j_{wy}}$ are constant. In fact, their value and their variation change for each object, since they depend on the value of several moments. That is why we will see, in Section V, how to choose for each object the best pair (c_i, c_j) from the set given in (9).

Finally, we recall that the interaction matrices (20) and (22) have nice expected forms only when the object is parallel to the image plane. In Section V, we will see that when the desired object position is parallel to the image plane, satisfactory results are obtained, even if the initial position is far away from this configuration. However, if the desired object position is not parallel to the image plane, the decoupling and linearizing properties are not as good as in the parallel case. That is why we present, in the next section, a new method to generalize our results to the case where the desired object position may have any orientation with respect to the image plane (except, of course, the degenerate case where the camera optical center belongs to the object plane).

IV. Generalization to Desired Object Poses Non-Parallel to the Image Plane

The general idea of our method consists of applying a virtual rotation to the camera, computing the visual features after this virtual motion, and then using the transformed features in the control law. The rotation is determined so that the image plane in its virtual desired position is parallel to the object. The properties obtained in this case will thus be enlarged for any desired configuration.

The first step of the method consists of determining the virtual rotation $\mathbf{R}^*$ to apply to the camera. If the task is specified by a desired configuration to reach between the camera and the object, $\mathbf{R}^*$ is directly given by this configuration (but this method necessitates the knowledge of the 3-D model of the object to compute the desired value $\mathbf{s}^*$ of the visual features). If the task is specified by a desired image acquired during an offline learning step, $\mathbf{R}^*$ can be obtained either from a pose-estimation algorithm if the model of the object is known, or from a partial pose-estimation algorithm if another image of the object is available [29]. Finally, the two angles involved in $\mathbf{R}^*$ can also be given during the learning step in the same way as the desired depth Z^* is set for the parallel case. For this method, no prior knowledge of the pattern lying on the target plane is required. We have chosen this last simple solution for the experimental results presented in Section V. We will see that a coarse approximation of $\mathbf{R}^*$ is sufficient, since the decoupling properties are ensured in a neighborhood of the parallel configuration.

We now describe how the visual features are computed. Let us denote (X_t, Y_t, Z_t), and (X, Y, Z) the coordinates of a 3-D point after and before the virtual rotation. Of course, we have

$$\begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} = \mathbf{R}^* \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{23}$$

from which we immediately deduce the coordinates (x_t, y_t) of the point in the image that would have been obtained if the camera had really moved. Indeed, using the perspective projection equation ($x_t = X_t/Z_t$, $y_t = Y_t/Z_t$), we obtain

$$\begin{cases} x_t = \dfrac{(r_{11}x + r_{12}y + r_{13})}{(r_{31}x + r_{32}y + r_{33})} \\ y_t = \dfrac{(r_{21}x + r_{22}y + r_{23})}{(r_{31}x + r_{32}y + r_{33})} \end{cases} \tag{24}$$

where (x, y) are the coordinates of the point in the real image. We can note that (x_t, y_t) can be computed directly from $\mathbf{R}^*$ and (x, y). An estimation of the coordinates of the 3-D point is, thus, useless.

If the object is composed of a set of n points, (24) is applied to all the n points in the desired image (from which, visual features $\mathbf{s}_t^*$ are computed), and for all the n points in the current image (from which, visual features $\mathbf{s}_t$ are computed). Otherwise, if a dense object is considered, the moments after the virtual rotation are given by

$$m_{t_{pq}} = \iint_{\mathcal{O}_t} x_t^p y_t^q dx_t dy_t = \iint_{\mathcal{O}} x_t^p y_t^q |\mathbf{J_t}| dx dy \tag{25}$$

where

$$|\mathbf{J_t}| = \begin{vmatrix} \frac{\partial x_t}{\partial x} & \frac{\partial x_t}{\partial y} \\ \frac{\partial y_t}{\partial x} & \frac{\partial y_t}{\partial y} \end{vmatrix} = \frac{1}{(r_{31}x + r_{32}y + r_{33})^3}.$$

We thus obtain

$$m_{t_{pq}} = \iint_{\mathcal{O}} \frac{(r_{11}x + r_{12}y + r_{13})^p (r_{21}x + r_{22}y + r_{23})^q}{(r_{31}x + r_{32}y + r_{33})^\gamma} dx dy \tag{26}$$

where $\gamma = p+q+3$. Computing the moments directly from (26) is possible, but time consuming. We thus propose a more efficient method, based on a Taylor series expansion of $1/(r_{31}x + r_{32}y + r_{33})^\gamma$. Indeed, if $r_{31}x + r_{32}y \ll r_{33}$ (which is the case, in practice), this term can be approximated by

$$\frac{1}{(r_{31}x + r_{32}y + r_{33})^\gamma} \approx \frac{1}{r_{33}^\gamma} \left(1 - \gamma \frac{(r_{31}x + r_{32}y)}{r_{33}} + \dots\right). \tag{27}$$

$$\begin{cases} \alpha_{wx} = \dfrac{\left(\beta\left[\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21})\right] + \gamma x_g\left[\mu_{02}(\mu_{20} - \mu_{02}) - 2\mu_{11}^2\right] + \gamma y_g \mu_{11}[\mu_{20} + \mu_{02}]\right)}{d} \\ \alpha_{wy} = \dfrac{\left(\beta\left[\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12})\right] + \gamma x_g \mu_{11}[\mu_{20} + \mu_{02}] + \gamma\left[\mu_{20}(\mu_{02} - \mu_{20}) - 2\mu_{11}^2\right]\right)}{d}, \quad d = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \end{cases}$$

Using (27) in (26), and after simple developments, we obtain

$$m_{t_{pq}} \approx \frac{r_{13}^p r_{23}^q}{r_{33}^\gamma} \sum_{k_1=0}^{p} \sum_{l_1=0}^{k_1} \sum_{k_2=0}^{q} \sum_{l_2=0}^{k_2} \binom{k_1}{p} \binom{l_1}{k_1} \binom{k_2}{q} \binom{l_2}{k_2} \times \left(\frac{r_{11}}{r_{13}}\right)^{l_1} \left(\frac{r_{12}}{r_{13}}\right)^{k_1-l_1} \left(\frac{r_{21}}{r_{23}}\right)^{l_2} \left(\frac{r_{22}}{r_{23}}\right)^{k_2-l_2} \times \left(m_{l,k-l} - \frac{(r_{31} m_{l+1,k-l} + r_{32} m_{l,k-l+1})}{r_{33}} + \dots\right) \tag{28}$$

with $k = k_1 + k_2$ and $l = l_1 + l_2$. The moments after the virtual rotation can thus be computed directly and efficiently from the moments in the real image.

Finally, a change in the control law (2) has to be performed to come back from the virtually rotated camera to the real one

$$\mathbf{v} = -\lambda \mathbf{V} \widehat{\mathbf{L}_\mathbf{s}}^{-1} (\mathbf{s}_t - \mathbf{s}_\mathbf{t}^*) \tag{29}$$

where matrix $\mathbf{V}$ is nothing but

$$\mathbf{V} = \begin{bmatrix} \mathbf{R}^{*\top} & 0 \\ 0 & \mathbf{R}^{*\top} \end{bmatrix}.$$

Let us note that $\mathbf{V}$ is a constant block-diagonal matrix, and thus preserves the decoupling properties between translational and rotational motions. Let us also note that the virtual rotation does not change the stability properties of the system, even if it is coarsely approximated, as long as we avoid the degenerate case where the camera optical center belongs to the object plane (which is very unlikely to occur).

V. Experimental Results

This section presents several experimental results obtained at video rate with a 6-DOF eye-in-hand system using first dense objects, and then discrete ones. For dense objects, image moments are efficiently computed from the contour points using Green's theorem [26]. In all the experiments, and as in [28], we have used the following model of the interaction matrix in the control law (29):

$$\widehat{\mathbf{L}_\mathbf{s}} = \frac{1}{2}\left(\mathbf{L}^{\|}_{\mathbf{s}(\mathbf{s}_\mathbf{t}^*)} + \mathbf{L}^{\|}_{\mathbf{s}(\mathbf{s}_\mathbf{t})}\right).$$

Indeed, it has been recently proved in [17] that this choice largely improves the system behavior. Note that, of course, the improvements appear only for the terms of $\mathbf{L}_\mathbf{s}$ which are not constant.

A. Experimental Results Using Dense Objects

We first consider the object depicted in Fig. 1(a) and (b): a "whale." For this object, we have chosen c_9 and c_{10} from the set given in (9) to control the rotational motion ω_x and ω_y. For all the possible pairs, we can indeed compute, using (28), the error

$$e_c(\alpha, \beta) = \left(c_{i_t}(\alpha, \beta) - c_{i_t}^*\right)^2 + \left(c_{j_t}(\alpha, \beta) - c_{j_t}^*\right)^2 \tag{30}$$

where α and β, which represent the rotation angles around the x and y axes, are varying from $-(\pi/3)$ to $\pi/3$, typically. We then choose the pair (c_i, c_j) such that the error e_c presents a global minimum, with an influence zone as large and as symmetrical as possible [31]. A complete study is given in [27]. We present in Fig. 2 the value of e_c for four possible pairs. We can note from

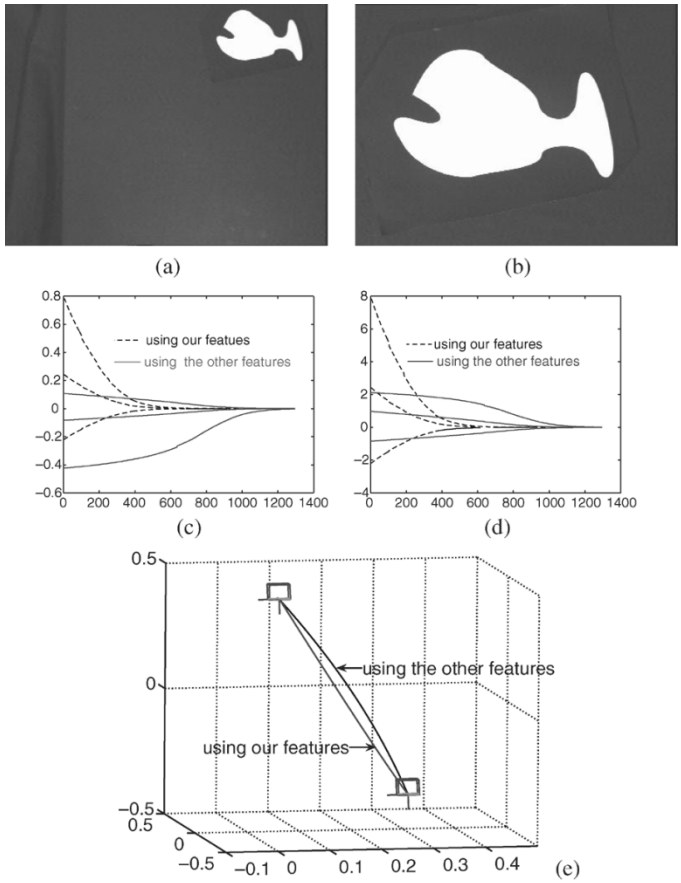

Fig. 1. Results for a pure translational motion when the object is parallel to the image plane. (a) Initial image. (b) Desired image. (c) $\mathbf{s} - \mathbf{s}^*$ m. (d) $\boldsymbol{v}_\mathbf{c}$ cm/s. (e) Camera 3-D trajectory.

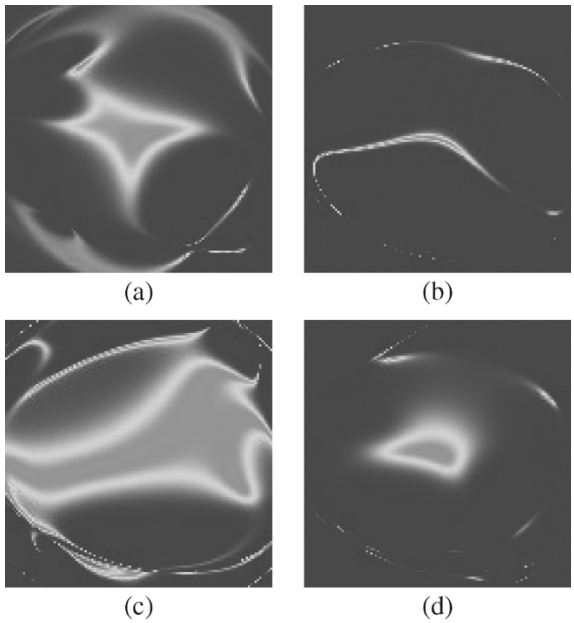

Fig. 2. Representation of $f(e_c)$ on $[-(\pi/3); \pi/3] \times [-(\pi/3); \pi/3]$ for the "whale" and pair. (a) (c_9, c_{10}). (b) (c_6, c_4). (c) (c_3, c_4). (d) (c_9, c_5).

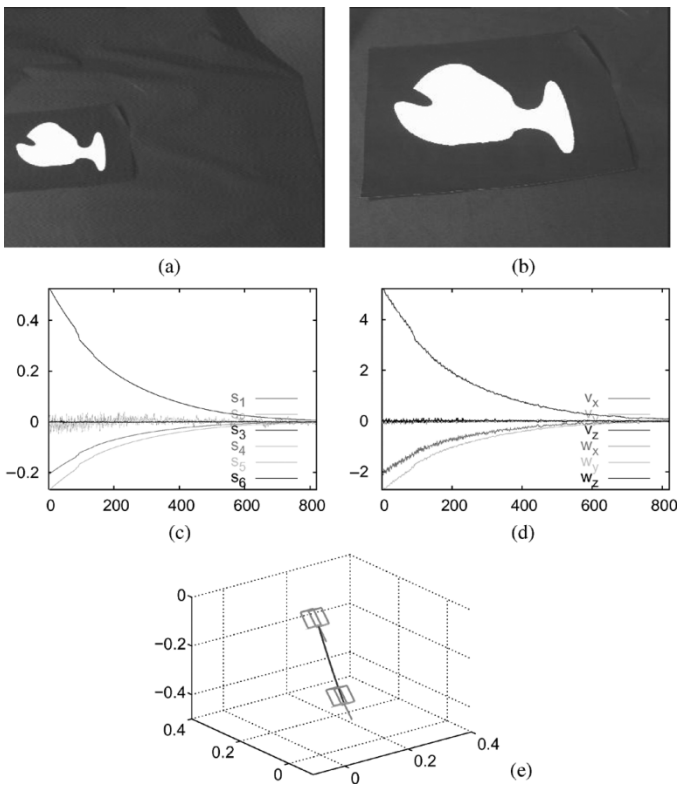

Fig. 3. Results for a pure translation when the object is not parallel to the image plane. (a) Initial image. (b) Desired image. (c) $\mathbf{s} - \mathbf{s}^*$ m. (d) $\mathbf{v_c}$ cm/s and dg/s. (e) Camera 3-D trajectory.

Fig. 2(a) and (d) that (c_9, c_{10}) and (c_9, c_5) supply an adequate behavior of e_c (that is why (c_9, c_{10}) has been selected). On the other hand, (c_6, c_4) and (c_3, c_4) have to be avoided [see Fig. 2(b) and (c)]. This offline selection process has to be done for each new object considered, once a desired image is acquired. Computing the values of $e_c(\alpha, \beta)$ for all the possible pairs (c_i, c_j) currently takes a few seconds on a standard PC.

In the next paragraph, we consider the case where a pure translational motion can be realized to reach the desired image from the initial one. We will then consider more complex displacements.

1) Pure Translational Motion: We compare in Fig. 1 the results obtained with our features given in (19) and those obtained using the centroid coordinates (x_g, y_g) and the area a. For that first experiment, the image plane is parallel to the object plane, the desired depth Z^* has been set to 0.5 m, and gain λ to 0.1. We can see in Fig. 1 the improvements brought by the proposed features (corresponding plots are in dashed lines), since they allow obtaining the same exponential decoupled decrease for the visual features and for the components of the camera velocity. As expected, the camera 3-D trajectory is a pure straight line using the proposed features, while it is not using the other ones. Note that using a bad estimation of Z^* with our features just has a gain effect. Thus, it changes the time to convergence of the system, but not the robot trajectory nor the exponential decoupled decrease of the features (as long as $\widehat{Z^*} > 0$).

The results still remain good when the image and object planes are not parallel (see Fig. 3). In that case, Z^* has again been set to 0.5 m, and rotation $\mathbf{R}^*$ has been specified, by

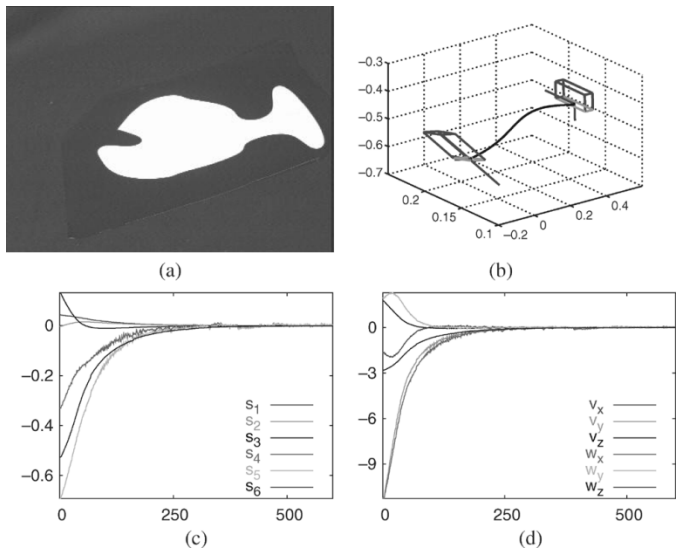

Fig. 4. Results for a complex motion when the desired object position is parallel to the image plane. (a) Initial image. (b) Camera 3-D trajectory. (c) $\mathbf{s} - \mathbf{s}^*$ m. (d) $\mathbf{v_c}$ cm/s and dg/s.

hand, as a rotation of 30° around the x axis. The camera 3-D trajectory is still a straight line, while if the virtual rotation is not applied, rotational motions are involved to reach the goal, whatever the visual features used. This validates the selected features to control the translational motions, and also the importance of applying a virtual rotation when the desired object position is not parallel to the image plane.

2) Complex Motion: We now test our scheme for displacements involving very large translations and rotations. We first consider the case where the image and object planes are parallel at the desired position. The desired image is depicted in Fig. 1(b), while the initial one is in Fig. 4(a). The numerical value of the interaction matrix computed for the desired position is given by

$$\mathbf{L}^{\parallel}_{\mathbf{s}(\mathbf{s}^*)} = \begin{bmatrix} -1 & 0 & 0 & 0.01 & -0.52 & 0.01 \\ 0 & -1 & 0 & 0.51 & -0.01 & 0.01 \\ 0 & 0 & -1 & -0.02 & -0.01 & 0 \\ 0 & 0 & 0 & -0.61 & 0.09 & 0 \\ 0 & 0 & 0 & -0.33 & -0.62 & 0 \\ 0 & 0 & 0 & -0.04 & -0.08 & -1 \end{bmatrix}.$$

As expected, we can note that $\mathbf{L}^{\parallel}_{\mathbf{s}(\mathbf{s}^*)}$ is block triangular with main terms around the diagonal. Its condition number, 2.60, is very satisfactory. The obtained results are given in Fig. 4. They show the good behavior of the control law. Indeed, there is no oscillation in the decreasing of the visual features [see Fig. 4(c)], and there is only one small oscillation for only two components of the camera velocity [see Fig. 4(d)]. Even if the rotation to realize between the initial and the desired positions is very large, the obtained camera 3-D trajectory is satisfactory [see Fig. 4(b)]. The decoupling properties are thus ensured in a large neighborhood of the desired position, despite the fact that the current object plane parameters are never computed nor introduced in the control law. The behavior obtained with our pure image-based method is thus similar to that obtained with a hybrid control scheme (but it does not necessitate an estimation of the camera

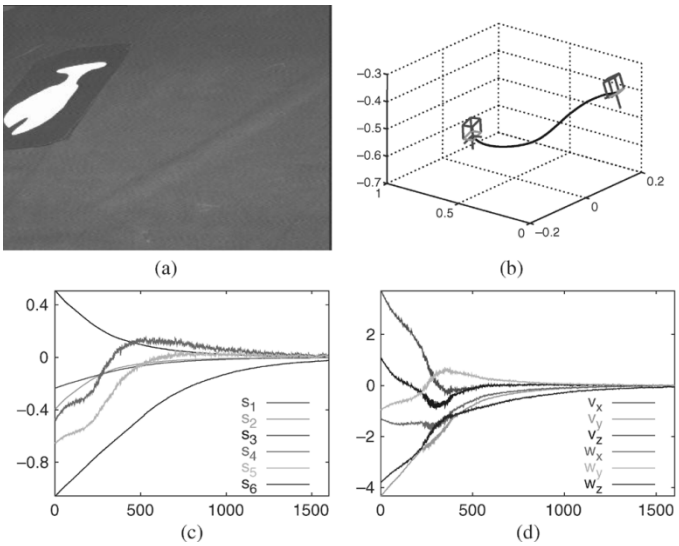

Fig. 5. Results for a complex motion when the desired object position is not parallel to the image plane. (a) Initial image. (b) Camera 3-D trajectory. (c) $\mathbf{s} - \mathbf{s}^*$ m. (d) $\mathbf{v_c}$ cm/s and dg/s.

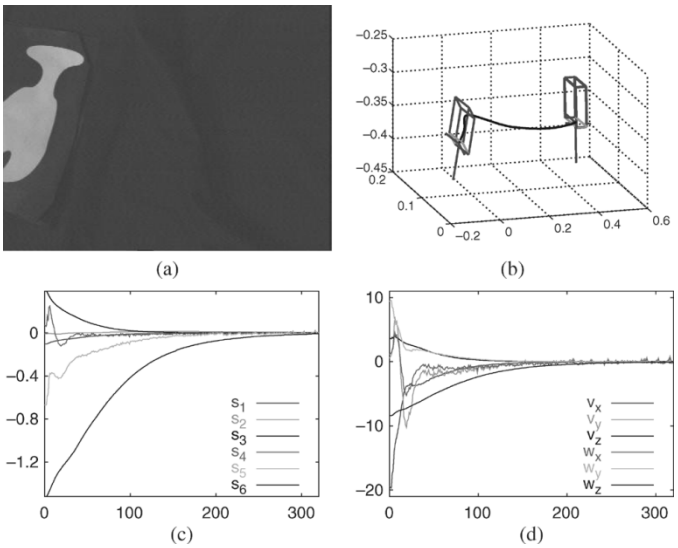

Fig. 6. Results using a bad calibration. (a) Initial image. (b) Camera 3-D trajectory. (c) $\mathbf{s} - \mathbf{s}^*$ m. (d) $\mathbf{v_c}$ cm/s and dg/s.

displacement at each iteration of the control scheme). We can finally note that using moments of order 5 (involved in c_9 and c_{10}) does not introduce significant noise in the control law.

The results obtained when the image and the object planes are not parallel for the desired position are given in Fig. 5. We can note the very large difference between the initial image [Fig. 5(a)] and the desired one [Fig. 3(b)]. Thanks to the virtual rotation applied and the visual features selected, the camera trajectory is still very satisfactory, as well as the decreasing of the visual features and the components of the camera velocity.

3) Results With a Bad Calibration and Object Occlusion: We now test the robustness of our approach with respect to a bad calibration of the system. In the experiment reported in Fig. 6, errors have been added to camera intrinsic parameters

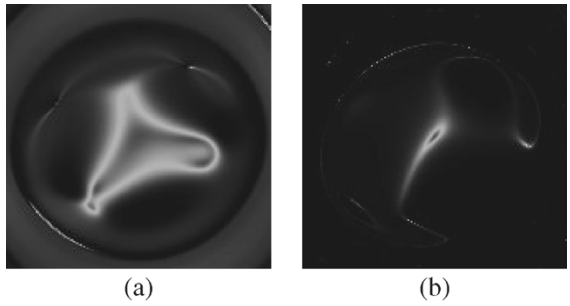

Fig. 7. Representation of $f(e_c)$ for the "brain" and for the pair. (a) (c_9, c_{10}). (b) (c_6, c_4).

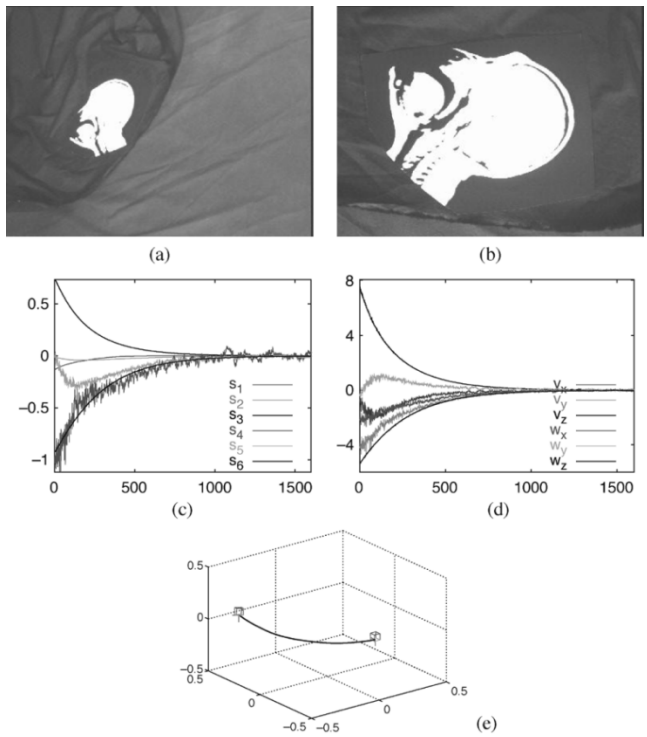

Fig. 8. Results obtained with the "brain" for a desired object position parallel to the image plane. (a) Initial image. (b) Desired image. (c) $\mathbf{s} - \mathbf{s}^*$ m. (d) $\mathbf{v_c}$ cm/s and dg/s. (e) Camera 3-D trajectory.

(25% on the focal length and 20 pixels on the coordinates of the principal point) and to the object plane parameters ($\widehat{Z}^* = 0.8$ m, instead of $Z^* = 0.5$ m). The lighting conditions from the initial position [see Fig. 6(a)] to the desired one [see Fig. 1(b)] are also different. Furthermore, an occlusion has been generated, since the object is not completely in the camera FOV at the beginning of the servo. Despite the worse conditions of the experiments, the system still converges, and, as soon as the occlusion ends (after iteration 30), the behavior of the system is similar to that of the previous experiments, which validates the robustness of our scheme with respect to modeling errors.

4) Results for Another Object: We now consider another object, a "brain" [see Fig. 8(a) and (b)]. For that object, pair (c_6, c_4) has been selected to control the rotational motion ω_x and ω_y. Indeed, it is clear from Fig. 7 that this pair is now adequate, while (c_9, c_{10}) is not. The numerical value of the interaction matrix

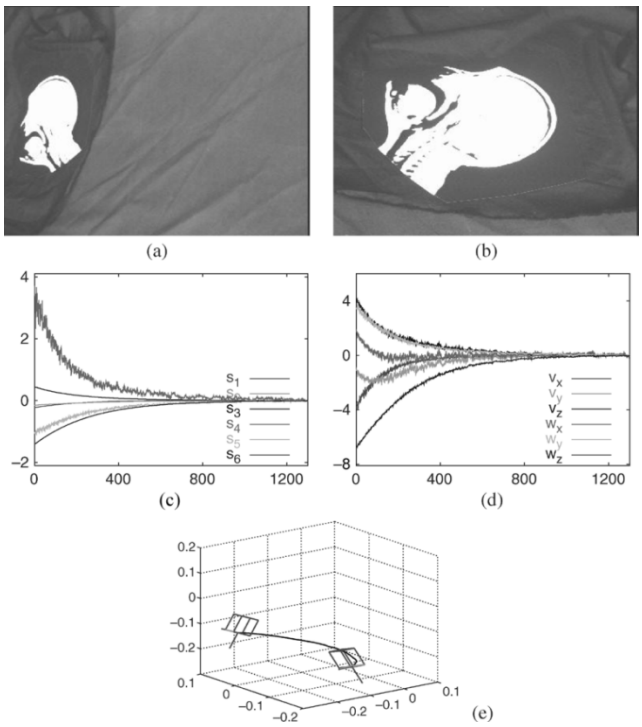

Fig. 9. Results obtained with the "brain" for a desired object position non-parallel to the image plane. (a) Initial image. (b) Desired image. (c) Visual features $\mathbf{s} - \mathbf{s}^*$ m. (d) $\mathbf{v_c}$ cm/s and dg/s. (e) Camera 3-D trajectory.

corresponding to the desired configuration given in Fig. 8(b) is given by

$$\mathbf{L}^{\parallel}_{\mathbf{s}(\mathbf{s}^*)} = \begin{bmatrix} -1 & 0 & 0 & -0.00 & -0.51 & 0.00 \\ 0 & -1 & 0 & 0.51 & 0.00 & 0.01 \\ 0 & 0 & -1 & -0.00 & -0.02 & 0 \\ 0 & 0 & 0 & -3.38 & -2.68 & 0 \\ 0 & 0 & 0 & -0.90 & 1.03 & 0 \\ 0 & 0 & 0 & -0.10 & 0.00 & -1 \end{bmatrix}.$$

Once again, we can note the nice form of $\mathbf{L}^{\parallel}_{\mathbf{s}(\mathbf{s}^*)}$ whose condition number is 3.14. We can also note in Fig. 8(c)–(e) the correct behavior of the control law, which is very similar to the one obtained with the "whale."

Similarly, if we consider a desired configuration of the object plane non-parallel to the image plane, the behavior obtained is still satisfactory, thanks to the virtual rotation applied (see Fig. 9, where $\mathbf{R}^*$ has again been specified as a rotation of 30° around the x axis).

B. Experimental Results Using Discrete Objects

Discrete objects are now considered. The first one is very simple and composed of 17 "white dots" (see Fig. 11). For that object, several pairs of moments invariants can be chosen, such as (c_9, c_{10}) or (c_6, c_4) (see Fig. 10). In the following experiments, we have chosen the pair (c_6, c_4). Two cases for the desired camera position have been also considered. Either the image plane is parallel to the object, or it is not. The corresponding images are given in Fig. 11(a) and (b).

1) Pure Translational Motion: In this experiment, the same pure translation $T = (-24 \text{ cm}, 17 \text{ cm}, -70 \text{ cm})$ is between the initial and the desired configurations for both parallel and non-parallel cases [see Fig. 11(c) and (d)]. We have compared the results obtained using the moments proposed in Section III-A as visual features [see (19) and (21)], and using all the points coordinates (x_k, y_k).

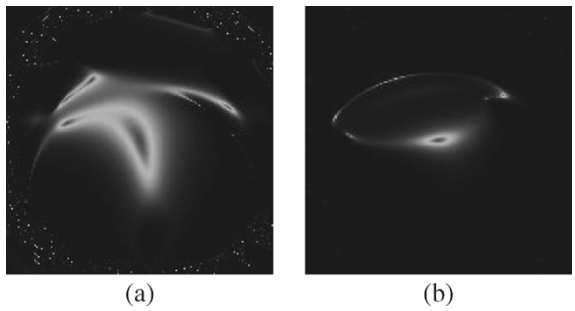

Fig. 10. Representation of $f(e_c)$ for the set of points and for the pair. (a) (c_9, c_{10}). (b) (c_6, c_4).

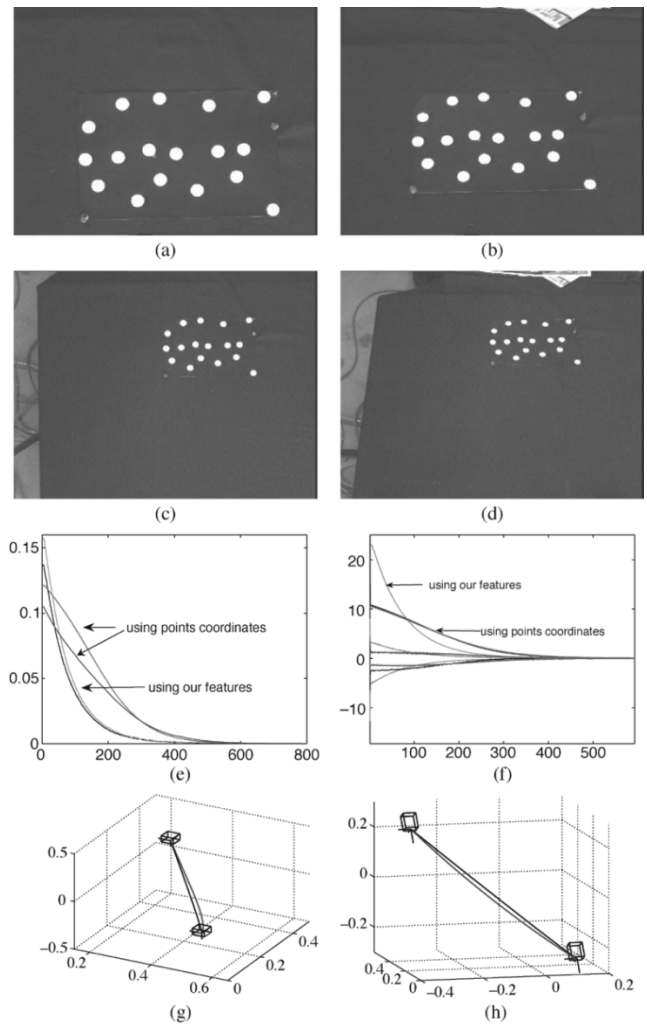

Fig. 11. Pure translational motion. (a) Desired image when the object is parallel to the image plane. (b) Desired image when the object is not parallel to the image plane. (c) Initial image for a pure translation from (a). (d) Initial image for a pure translation from (b). (e) Comparison of $\mathbf{s} - \mathbf{s}^*$ m. (f) Comparison of $\boldsymbol{v_c}$ cm/s. (g) Camera 3-D trajectory when the object is parallel to the image plane (a straight line, using our features). (h) Idem when the object is not parallel to the image plane.

In both parallel and non-parallel cases, we can see in Fig. 11(e)–(h) the improvements brought using moments and,

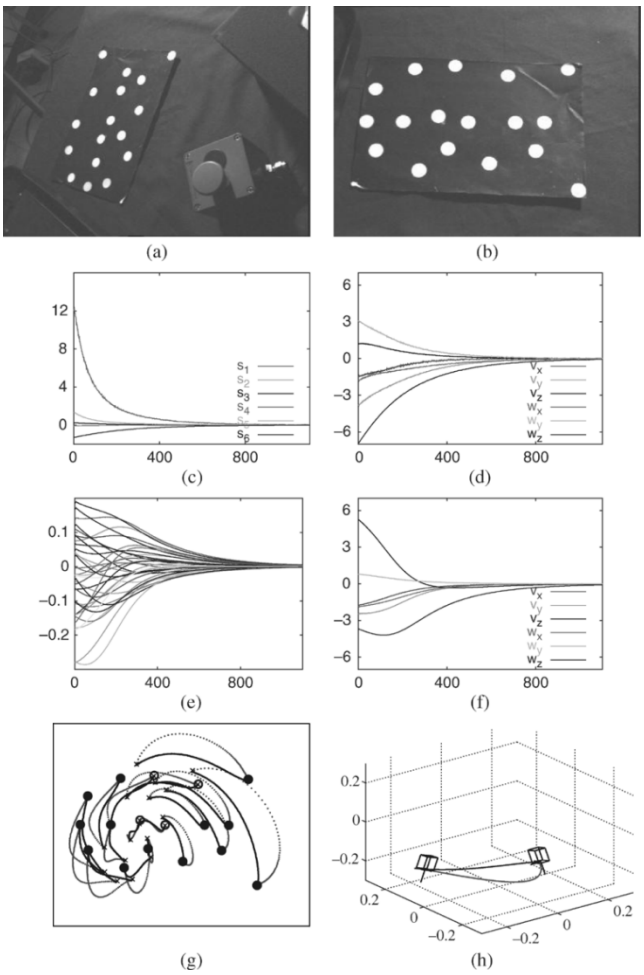

Fig. 12. Results for a complex motion using a discrete object, compared using our visual features and using a basic image-based approach. (a) Initial image. (b) Desired image. (c) $\mathbf{s} - \mathbf{s}^*$ m using moments. (d) $\mathbf{v_c}$ cm/s and dg/s using moments. (e) $\mathbf{s} - \mathbf{s}^*$ m using image-points coordinates. (f) $\mathbf{v_c}$ cm/s and dg/s using image-points coordinates. (g) Image points trajectories (with dashed lines using image points coordinates). (h) Camera trajectories (almost a straight line with our features).

for the non-parallel case, brought by the virtual rotation. Indeed, they allow obtaining a pure exponential decrease for the visual features, and generate exactly the same camera velocity. As expected, the camera 3-D trajectory is a pure straight line in both cases using the proposed method. When points coordinates are used, we no longer have a pure exponential decrease for the visual features and for the camera velocity components. The camera trajectory is thus no longer a straight line. Rotational motions (unfortunately, not visible on the presented plots) are even involved when points are used for the non-parallel case.

2) Complex Motion and Comparison wrt Basic Image-Based Visual Servoing: We now consider a complex displacement, and we present a comparison between our image-based method using six combinations of moments and the traditional image-based method, where the coordinates of the image points are used as input to the control scheme. The initial image is given in Fig. 12(a), while the desired one is given in Fig. 12(b). The corresponding displacement is very large ($t_x = -38$ cm, $t_y = 47$ cm, $t_z = 10$ cm, $\theta u_x = 23°$, $\theta u_y = -27°$, $\theta u_z = 64°$), and the desired position is such that there is a rotation of $20°$ around the

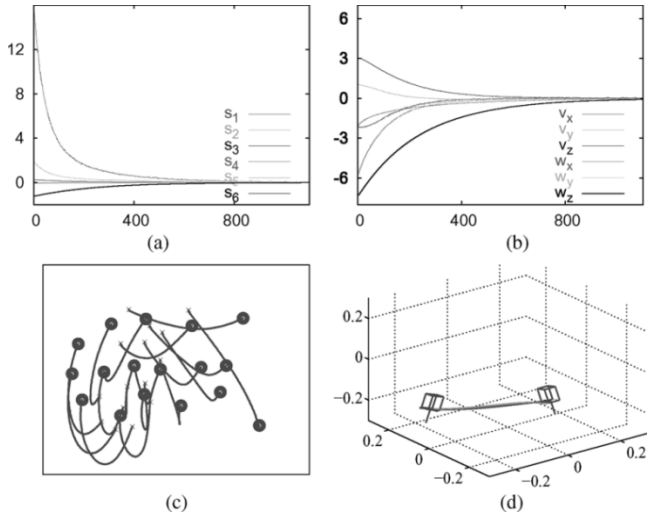

Fig. 13. Results with modeling errors. (a) $\mathbf{s} - \mathbf{s}^*$ m. (b) $\mathbf{v_c}$ cm/s and dg/s. (c) Image points trajectories. (d) Camera trajectory (almost the same, with or without modeling errors).

x camera axis between the object plane and the image plane. As for the other objects considered previously, the interaction matrix using the moments after the virtual rotation is sparse and block triangular

$$\mathbf{L}_{\mathbf{s}(\mathbf{s}_t^*)}^{\parallel} = \begin{bmatrix} -1 & 0 & 0 & 0.00 & -0.51 & 0.18 \\ 0 & -1 & 0 & 0.48 & -0.00 & 0.03 \\ 0 & 0 & -1 & -0.24 & -0.02 & 0 \\ 0 & 0 & 0 & 3.44 & -20.56 & 0.0 \\ 0 & 0 & 0 & 1.19 & -2.31 & 0.0 \\ 0 & 0 & 0 & 0.14 & -0.72 & -1 \end{bmatrix}.$$

Despite the large displacement to realize, we can note in Fig. 12(c) and (d) the decoupled and exponential decrease of the six combinations of moments and the six camera velocity components. If points coordinates are used as input to the control law [see Fig. 12(e) and (f)], the system still converges, but without these nice properties. The difference of behaviors is particularly clear on the trajectories of the image points [see Fig. 12(g)] and the camera trajectory [see Fig. 12(h)], where our scheme leads to almost a pure 3-D straight line. The results obtained using the proposed combinations of 2-D moments are thus similar to those of a hybrid visual servoing. Even if currently limited to planar objects, our method has the advantage that it is not necessary to explicitly solve the matching problem between each point of the object extracted on the current image, and the corresponding point extracted on the desired image. Contrary to the other existing methods (that is, basic image-based, position-based, and hybrid schemes), it is sufficient to check that the same set of points is used to compute the moments in the current and the desired images.

3) Results With a Bad System Calibration: We now test the robustness of our approach with respect to modeling errors. In the presented experiment, errors have been added to camera intrinsic parameters (25% on the focal length and 20 pixels on the coordinates of the principal point) and to the object depth ($\widehat{Z}^* = 1$ m instead of $Z^* = 0.7$ m). Furthermore, an error equal to $10°$ has been set in $\mathbf{R}^*$. The results are given in Fig. 13. Even if the trajectory of the points are different in the calibrated

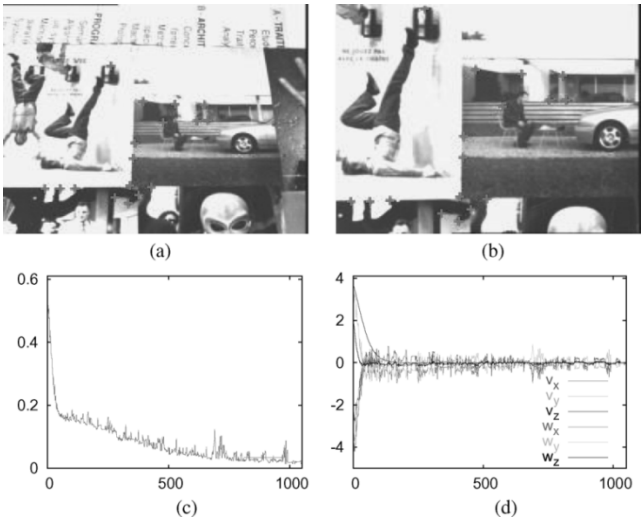

Fig. 14. Results for complex images. (a) Initial image. (b) Desired image. (c) $\|\mathbf{s}-\mathbf{s}^*\|$ m. (d) $\mathbf{v_c}$ cm/s and dg/s.

and coarse-calibrated cases [compare Figs. 12(g) and 13(c)], which is mainly due to the large errors introduced in the camera intrinsic parameters, all the errors introduced have a small effect on the decreasing of the moments [compare Figs. 12(c) and 13(a)], on the camera velocity components [compare Figs. 12(d) and 13(b)], and thus on the camera 3-D trajectory, which is still very near to a pure straight line, as in the calibrated case [see Fig. 13(d), where the trajectories can be compared]. Let us note that the basic image-based method using points coordinates as input to the control law does not allow the system to converge with the same modeling errors, since some points leave the camera FOV. These results validate the strong robustness of our scheme with respect to modeling errors, and the fact that the decoupling properties are not sensitive to a coarse approximation of the virtual rotation, since they are ensured in a neighborhood of the virtually rotated desired position.

4) Results for Complex Images: Finally, we present experimental results obtained with more complex images (see Fig. 14) using the pair (c_9, c_{10}). The considered points have been extracted using the well-known Harris detector and tracked using a simple sum of squared distances algorithm. We can note that the plots obtained are more noisy than using simple dots, because of the less accurate points extraction. It is mainly noticeable on ω_x and ω_y components of the camera velocity, since these values depend on moments of order 5 (while ω_z and v_z are not noisy at all, since their value only depends on moments of order 2). Despite this noise, the exponential decrease, the convergence, and the stability are still obtained, which proves the validity of our approach. This results could be improved easily using a subpixel accuracy image tracker, such as the Shi-Tomasi algorithm [25].

VI. Conclusion

In this paper, we have proposed a new visual servoing scheme based on image moments, valid for dense and discrete objects. Six features have been designed to decouple the DOFs of the system, which provides a large domain of convergence, a good behavior of the visual features, and an adequate camera trajectory. A new method, based on a virtual camera rotation, has also been proposed to extend the decoupling properties for any desired camera orientation with respect to the object. The experimental results have shown the validity of our approach, and its robustness with respect to modeling errors. We have obtained results similar to using a hybrid visual servoing scheme, but with a more simple cooking of the visual features, since our method does not necessitate any point-to-point matching, any homography estimation, nor any partial displacement estimation at each iteration of the control scheme. Future work will be devoted to determining a unique pair of moments invariants able to efficiently control ω_x and ω_y, whatever the considered object. We would like also to generalize the results to nonplanar objects, and to determine analytical conditions to ensure the global stability of the system in the presence of modeling errors.

Appendix

The set of invariants involved in (9) are given by [27]

$$
\begin{aligned}
I_1 &= -\mu_{20}\mu_{02}+\mu_{11}^2\\
I_2 &= (\mu_{20}-\mu_{02})^2+4\mu_{11}^2\\
I_3 &= (\mu_{30}-3\mu_{12})^2+(3\mu_{21}-\mu_{03})^2\\
I_4 &= (\mu_{30}+\mu_{12})^2+(\mu_{21}+\mu_{03})^2\\
I_5 &= -\mu_{30}^2\mu_{03}^2+6\mu_{30}\mu_{21}^3-4\mu_{30}\mu_{12}^3-4\mu_{21}^3\mu_{03}\\
&\quad +3\mu_{21}^2\mu_{12}^2\\
I_6 &= 3\mu_{30}^2\mu_{12}^2+2\mu_{30}^2\mu_{03}^2-6\mu_{30}\mu_{21}^2\mu_{12}-6\mu_{30}\mu_{21}\mu_{12}\mu_{03}\\
&\quad +2\mu_{30}\mu_{12}^3+3\mu_{21}^4+2\mu_{21}^3\mu_{03}+3\mu_{21}^2\mu_{03}^2\\
&\quad -6\mu_{21}\mu_{12}^2\mu_{03}+3\mu_{12}^4-6\mu_{21}\mu_{12}^2\mu_{03}+3\mu_{12}^4\\
I_7 &= -\mu_{30}^3\mu_{03}+3\mu_{30}^2\mu_{21}\mu_{12}-2\mu_{30}\mu_{21}^3-3\mu_{30}\mu_{21}^2\mu_{03}\\
&\quad +6\mu_{30}\mu_{21}\mu_{12}^2+3\mu_{30}\mu_{12}^2\mu_{03}+\mu_{30}\mu_{03}^3-3\mu_{21}^3\mu_{12}\\
&\quad -6\mu_{21}^2\mu_{12}\mu_{03}+3\mu_{21}\mu_{12}^3-3\mu_{21}\mu_{12}\mu_{03}^2+2\mu_{12}^3\mu_{03}\\
I_8 &= -\mu_{30}^3\mu_{12}+\mu_{30}^2\mu_{21}^2-\mu_{30}^2\mu_{21}\mu_{03}-2\mu_{30}^2\mu_{12}^2\\
&\quad +3\mu_{30}\mu_{21}^2\mu_{12}-6\mu_{30}\mu_{21}\mu_{12}\mu_{03}+3\mu_{30}\mu_{12}^3\\
&\quad -\mu_{30}\mu_{12}\mu_{03}^2+3\mu_{21}^3\mu_{03}+3\mu_{21}^3\mu_{03}-2\mu_{21}^2\mu_{03}^2\\
&\quad +3\mu_{21}\mu_{12}^2\mu_{03}-\mu_{21}\mu_{03}^3+\mu_{12}^2\mu_{03}^2\\
I_9 &= \mu_{30}^4+6\mu_{30}^3\mu_{12}+6\mu_{30}^2\mu_{21}\mu_{03}+9\mu_{30}^2\mu_{12}^2+2\mu_{30}^2\mu_{03}^2\\
&\quad +18\mu_{30}\mu_{21}\mu_{12}\mu_{03}+6\mu_{30}\mu_{12}\mu_{03}^2+9\mu_{21}^2\mu_{03}^2\\
&\quad +6\mu_{21}\mu_{03}^3+\mu_{03}^4\\
I_{10} &= \mu_{40}\mu_{04}-4\mu_{31}\mu_{13}+3\mu_{22}^2\\
I_{11} &= 3\mu_{40}\mu_{22}-2\mu_{40}\mu_{04}+3\mu_{31}^2+2\mu_{31}\mu_{13}-3\mu_{22}\mu_{04}\\
&\quad +3\mu_{13}^2\\
I_{12} &= 3\mu_{40}^2+12\mu_{40}\mu_{22}+2\mu_{40}\mu_{04}+16\mu_{31}\mu_{13}\\
&\quad +12\mu_{22}\mu_{04}+3\mu_{04}^2\\
I_{13} &= (\mu_{50}+2\mu_{32}+\mu_{14})^2+(\mu_{05}+2\mu_{23}+\mu_{41})^2\\
I_{14} &= (\mu_{50}-2\mu_{32}-3\mu_{14})^2+(\mu_{05}-2\mu_{23}-3\mu_{41})^2\\
I_{15} &= (\mu_{50}-10\mu_{32}+5\mu_{14})^2+(\mu_{05}-10\mu_{23}+5\mu_{41})^2.
\end{aligned}
$$

References

[1] Y. S. Abu-Mustapha and D. Psaltis, "Image normalization by complex moments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 1, pp. 46–55, Jan. 1985.

[2] S. O. Belkassim, M. Shridhar, and M. Ahmadi, "Shape-contour recognition using moment invariants," in *Proc. 10th Int. Conf. Pattern Recog.*, Atlantic City, NJ, Jun. 1990, pp. 649–651.
[3] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*. New York: Springer-Verlag, 1998, vol. 237, pp. 66–78.
[4] ——, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.
[5] C. Collewet and F. Chaumette, "A contour approach for image-based control of objects with complex shape," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Takamatsu, Japan, Nov. 2000, pp. 751–756.
[6] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robot. Autom.*, vol. 17, no. 4, pp. 507–515, Aug. 2001.
[7] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 521–533, Aug. 2002.
[8] K. Deguchi, "A direct interpretation of dynamic images with camera and object motions for vision-guided robot control," *Int. J. Computer Vision*, vol. 37, no. 1, pp. 7–20, Jun. 2000.
[9] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.
[10] J. Flusser, "On the independance of rotation moment invariants," *Pattern Recog.*, vol. 33, pp. 1405–1410, 2000.
[11] T. Hamel and R. Mahony, "Visual servoing of an underactuated dynamic rigid body system: An image-based approach," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 187–198, Apr. 2002.
[12] M.-K. Hu, "Visual pattern recognition by moments invariants," *IRE Trans. Inf. Theory*, vol. 8, pp. 179–187, Feb. 1962.
[13] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
[14] M. Iwatsuki and N. Okiyama, "A new formulation of visual servoing based on cylindrical coordinates system," *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 266–273, Apr. 2005.
[15] J.-S. Lee, I. H. Suh, B.-J. You, and S.-R. Oh, "A novel visual servoing approach involving disturbance observer," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Detroit, MI, May 1999, pp. 269–274.
[16] R. Mahony, P. Corke, and F. Chaumette, "Choice of image features for depth-axis control in image-based visual servo control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, Oct. 2002, pp. 390–395.
[17] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, Apr. 2004, pp. 1843–1848.
[18] E. Malis, F. Chaumette, and S. Boudet, "2-1/2 D visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.
[19] A. G. Mamistvalov, "n-dimensional moment invariants and conceptual theory of recognition n-dimensional solids," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 8, pp. 819–831, Aug. 1998.
[20] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 534–549, Aug. 2002.
[21] R. Mukundan and K. R. Ramakrishnan, *Moment Functions in Image Analysis Theory and Application*, Singapore: World Scientific, 1998.
[22] R. J. Prokop and A. P. Reeves, "A survey of moments-based techniques for unoccluded object representation," *Graph. Models Image Process.*, vol. 54, no. 5, pp. 438–460, Sep. 1992.
[23] S. S. Reddi, "Radial and angular moment invariants for image identification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, no. 2, pp. 240–242, Feb. 1981.
[24] P. Rives and J. Azinheira, "Linear structures following by an airship using vanishing points and horizon line in a visual servoing scheme," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, New Orleans, LA, Apr. 2004, pp. 255–260.
[25] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Int. Conf. Computer Vis. Pattern Recog.*, Seattle, WA, Jun. 1994, pp. 593–600.
[26] C. Steger, "On the calculation of arbitrary moments of polygons," Munchen Univ., Munchen, Germany, Tech. Rep. FGBV-96-05, Oct. 1996.
[27] O. Tahri, "Application des moments à l'asservissement visuel et au calcul de pose," Ph.D. dissertation (in French), Univ. de Rennes 1, IRISA, Rennes, France, Mar. 2004.
[28] O. Tahri and F. Chaumette, "Application of moment invariants to visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, Sep. 2003, pp. 4276–4281.
[29] ——, "Complex objects pose estimation based on image moment invariants," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 438–443.
[30] M. R. Teague, "Image analysis via the general theory of moments," *J. Opt. Soc. Amer.*, vol. 70, pp. 920–930, Aug. 1980.
[31] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1992.
[32] A. Walin and O. Kübler, "Complete sets of complex Zernike moments invariants and the role of the pseudo-invariants," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 11, pp. 1106–1110, Nov. 1995.
[33] W. Wilson, C. Hulls, and G. Bell, "Relative end-effector control using Cartesian position-based visual servoing," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 684–696, Oct. 1996.
[34] P. Zanne, G. Morel, and F. Plestan, "Robust 3D vision-based control and planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 5, New Orleans, LA, Apr. 2004, pp. 4423–4428.

Omar Tahri was born in Fez, Morocco, in 1976. He received the Master's degree in photonics, images and system control from Louis Pasteur University, Strasbourg, France, in 2000, and the Ph.D degree in computer science from the University of Rennes, Rennes, France, in 2004.

He is currently in a Postdoctoral position with CEA-LIST, Fontenay-aux-Roses, France. His research interests include robotics and computer vision, especially visual servoing.

François Chaumette (M'98) was born in Nantes, France, in 1963 and graduated from École Nationale Supérieure de Mécanique, Nantes, in 1987. He 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, where he is now "Directeur de Recherches" and head of the Lagadic Group. His research interests include robotics and computer vision, especially visual servoing and active perception.