Image moments: Generic descriptors for decoupled image-based visual servo

O. Tahri, François Chaumette

Image Moments: Generic Descriptors
for Decoupled Image-based Visual Servo

Omar Tahri[1], François Chaumette[2]
IRISA/INRIA Rennes, Campus de Beaulieu, 35 042 Rennes cedex, France
[1]Omar.Tahri@irisa.fr   [2]Francois.Chaumette@irisa.fr

Abstract— **Moments are generic and (usually) intuitive descriptors that can be computed from several kinds of objects defined either from closed contours (continuous object) or a set of points (discrete object). In this paper, we propose to use moments to design a decoupled image-based visual servo. The analytical form of the interaction matrix related to the moments computed for a discrete object is derived, and we show that it is different of the form obtained for a continuous object. 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.**

I. INTRODUCTION

Numerous visual servoing methods have been proposed to position a camera with respect to an object [5]. Recent developments are concerned with the selection or the transformation of visual data to design visual features that improve the behavior of the system [2], [6], [7], [9]. The objective of these works is to avoid the potential problems that may appear using basic techniques (typically reaching local minima or a task singularity) [1]. The way to design adequate visual features is directly linked to the modeling of their interaction with the robot motion, from which all control properties can be theoretically analyzed. If the interaction is too complex (i.e. highly non linear and coupled), the analysis becomes impossible and the behavior of the system is generally not satisfactory in difficult configurations where large displacements (especially rotational ones) have to be realized.

In our previous works [9], a decoupled control has been proposed using the image moments computed from a continuous objet. Six combinaitions of moments were proposed to obtain an adequate behavior. In this paper a generalization of those results to a discrete object defined by a rigid set of coplanar points, extracted and tracked in an image sequence is given. The goal of the visual servo is to control the robot motion so that these points reach a desired position in the image. In practice, when complex images are considered, these points are generally a set of points of interest extracted for example using the Harris's detector [4] and tracked using a SSD algorithm [3]. Instead of using the coordinates of all the image points as inputs in the control scheme, we propose to use some particular image moments. We will see in this paper that the interaction matrix of moments defined from a set of points is different from that obtained in the continuous case.

These differences impose a new selection of the visual features to design a decoupled control scheme.

Furthermore, up to now, the decoupling properties have been obtained only when the object is parallel to the image plane [9]. In this paper, we propose an original method to deal with the more general case where the desired object position may have any orientation with respect to the camera frame. The idea consists in applying a virtual camera motion and in using the transformed visual features in the control law. This result is valid either for a discrete objet either for a continuous objet.

In the next section, we determine the analytical form of the interaction matrix related to the moments computed from a set of points. In Section 3, we design six decoupled visual features to control the six degrees of freedom of the system when the camera is parallel to the image plane. In Section 4, we generalize these results to the case where the object may have any orientation with respect to the camera. Finally, experimental results using a continuous objet and discrete objects are given in Section 5 to validate the proposed theoretical results.

II. INTERACTION MATRIX OF MOMENTS COMPUTED FROM A SET OF IMAGE POINTS

It is well known that the moments related to a set of N image points are defined by:

$$m_{ij} = \sum_{k=1}^{N} x_k^i \, y_k^j \qquad (1)$$

By deriving the above equation, we obtain:

$$\dot{m}_{ij} = \sum_{h=1}^{N} (i x_k^{i-1} y_k^j \, \dot{x}_k + j x_k^i y_k^{j-1} \dot{y}_k) \qquad (2)$$

The velocity of any image point $\mathbf{x}_k = (x_k, y_k)$ is given by the classical equation:

$$\dot{\mathbf{x}}_k = \mathbf{L}_{\mathbf{x}_k} \mathbf{v} \qquad (3)$$

where $\mathbf{L}_{\mathbf{x}_k}$ is the interaction matrix related to $\mathbf{x}_k$ and $\mathbf{v} = (\upsilon_x, \upsilon_y, \upsilon_z, \omega_x, \omega_y, \omega_z)$ is the kinematic screw between the camera and the object. More precisely, we have:

$$\mathbf{L}_{\mathbf{x}_k} = \begin{bmatrix} -1/Z_k & 0 & x_k/Z_k & x_k y_k & -1-x_k^2 & y_k \\ 0 & -1/Z_k & y_k/Z_k & 1+y_k^2 & -x_k y_k & -x_k \end{bmatrix} \qquad (4)$$

where Z_k is the depth of the corresponding 3D point.

If all the points belongs to a plane, we can relate linearly the inverse of the depth of any 3D point to its image coordinates

(by just applying the perspective equation $x = X/Z, y = Y/Z$ to the plane equation):

$$\frac{1}{Z_k} = Ax_k + By_k + C \qquad (5)$$

Using (5) in (3), the velocity of $\mathbf{x_k}$ can be written as:

$$\begin{cases} \dot{x}_k &= -(Ax_k + By_k + C)v_x \\ &+ \; x_k(Ax_k + By_k + C)v_z \\ &+ \; x_k y_k \omega_x - (1 + x_k^2)\omega_y + y_k\omega_z \\ \dot{y}_k &= -(Ax_k + By_k + C)v_y \\ &+ \; y_k(Ax_k + By_k + C)v_z \\ &+ \; (1 + y_k^2)\omega_x - x_k y_k \omega_y - x_k\omega_z \end{cases} \qquad (6)$$

Finally, using (6) in (2), the interaction matrix $\mathbf{L}_{m_{ij}}$ related to m_{ij} can be determined and we obtain after simple developments:

$$\mathbf{L}_{m_{ij}} = \begin{bmatrix} m_{d_{vx}} & m_{d_{vy}} & m_{d_{vz}} & m_{d_{wx}} & m_{d_{wy}} & m_{d_{wz}} \end{bmatrix} \qquad (7)$$

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

We can note that the obtained interaction matrix is not exactly the same if we consider the moments defined from a set of points or defined by integration on an area in the image. In the continuous case, the coefficients of the interaction matrix are given by (see [9]):

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

They are obtained from the following formula [9]:

$$\dot{m}_{ij} = \iint_{\mathcal{D}} [\frac{\partial f}{\partial x}\dot{x} + \frac{\partial f}{\partial y}\dot{y} + f(x,y)(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y})]dxdy$$

where $f(x,y) = x^i y^j$ and where $\mathcal{D}$ is the image part where the object projects. We can see that the first two terms of the above equation correspond exactly to the two terms present in (2). On the other hand, the third term does not appear in (2), which explains the differences on the obtained analytical forms.

To illustrate these differences on a simple example, we can consider moment m_{00}. In the discrete case, m_{00} is nothing but the number N of tracked points. This number is of course invariant to any robot motion (if the image tracking does not loose any point) and we can check by setting $i = j = 0$ in (7) that all the terms of $L_{m_{00}}$ are indeed equal to zero. In the continuous case, m_{00} represents the area of the object, and general robot motion modifies the value of m_{00}, as can be checked for instance from m_{vz} or m_{ω_x}.

Similarly, if we consider the centered moments defined by:

$$\mu_{ij} = \sum_{k=1}^{N}(x_k - x_g)^i(y_k - y_g)^j \qquad (8)$$

we obtain after tedious developments the interaction matrix $\mathbf{L}_{\mu_{ij}}$ and we can check it is slightly different to that obtained in the continuous case (see [9]):

$$\mathbf{L}_{\mu_{ij}} = \begin{bmatrix} \mu_{d_{vx}} & \mu_{d_{vy}} & \mu_{d_{vz}} & \mu_{d_{wx}} & \mu_{d_{wy}} & \mu_{d_{wz}} \end{bmatrix} \qquad (9)$$

with :
$$\begin{cases} \mu_{d_{vx}} &= -iA\mu_{ij} - iB\mu_{i-1,j+1} \\ \mu_{d_{vy}} &= -jA\mu_{i+1,j-1} - jB\mu_{ij} \\ \mu_{d_{vz}} &= -A\mu_{wy} + B\mu_{wx} + (i+j)C\mu_{ij} \\ \mu_{d_{wx}} &= (i+j)\mu_{i,j+1} + ix_g\mu_{i-1,j+1} \\ & \quad + (i+2j)y_g\mu_{ij} - in_{11}\mu_{i-1,j} - jn_{02}\mu_{i,j-1} \\ \mu_{d_{wy}} &= -(i+j)\mu_{i+1,j} - (2i+j)x_g\mu_{ij} \\ & \quad - jy_g\mu_{i+1,j-1} + in_{20}\mu_{i-1,j} + jn_{11}\mu_{i,j-1} \\ \mu_{d_{wz}} &= i\mu_{i-1,j+1} - j\mu_{i+1,j-1} \end{cases}$$

where $n_{ij} = \mu_{ij}/m_{00}$. As in the continuous case, if the object is parallel to the image plane (i.e. $A = B = 0$), we can note from $\mu_{d_{vx}}$ and $\mu_{d_{vy}}$ that all centered moments are invariant with respect to translational motions parallel to the image plane ($\mu_{d_{vx}} = \mu_{d_{vy}} = 0$ when $A = B = 0$).

Since the interaction matrices related to moments are similar in both discrete and continuous cases, we will use in the next section the recent results proposed in [9] to design six combinaisons on image moments able to control the six degrees of freedom of the system. We will see however that one of these six features can not be exactly the same.

III. FEATURES SELECTION FOR VISUAL SERVOING

The main objective of visual features selection is to obtain a sparse 6×6 full rank interaction matrix that changes slowly around the desired camera pose (our dream being to obtain the identity matrix...). In this section, we consider that the desired position of the object is parallel to the image plane (i.e. $A = B = 0$). The more general case where the desired position of the object is not necessarily parallel to the image plane will be treated in the next section.

In [9], the following visual features had been proposed for the continuous case:

$$\mathbf{s} = (x_n, y_n, a_n, r_1, r_2, \alpha) \qquad (10)$$

where:
$$\begin{cases} x_n = a_n x_g \; , \; y_n = a_n y_g \; , \; a_n = Z^*\sqrt{\frac{a^*}{a}} \\ r_1 = \frac{I_{n_1}}{I_{n_3}} \; , \; r_2 = \frac{I_{n_2}}{I_{n_3}} \; , \; \alpha = \frac{1}{2}\arctan(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}) \end{cases}$$

$a(= m_{00})$ being the area of the object in the image, a^* its desired value, Z^* the desired depth of the object, α the orientation of the object in the image, and r_1 and r_2 two combinations of moments invariant to scale, 2D translation and 2D rotation. More precisely, we have:

$$\begin{cases} I_{n_1} = (\mu_{50} + 2\mu_{32} + \mu_{14})^2 + (\mu_{05} + 2\mu_{23} + \mu_{41})^2 \\ I_{n_2} = (\mu_{50} - 2\mu_{32} - 3\mu_{14})^2 + (\mu_{05} - 2\mu_{23} - 3\mu_{41})^2 \\ I_{n_3} = (\mu_{50} - 10\mu_{32} + 5\mu_{14})^2 + (\mu_{05} - 10\mu_{23} + 5\mu_{41})^2 \end{cases}$$

In the discrete case, all these features can be used but the area m_{00} since, as already stated, it always have the same constant value N whatever the robot pose. We thus propose to replace the area and its desired value by:

$$a = \mu_{20} + \mu_{02} \text{ and } a^* = \mu_{20}^* + \mu_{02}^* \tag{11}$$

By denoting $\mathbf{L}_{\mathbf{s}}^{\parallel}$ the value of the interaction matrix for all camera poses such that the image plane is parallel to the object, we obtain for the selected visual features $\mathbf{s}$ and for such configurations:

$$\mathbf{L}_{\mathbf{s}}^{\parallel} = \begin{bmatrix} -1 & 0 & 0 & a_n\epsilon_{11} & -a_n(1+\epsilon_{12}) & y_n \\ 0 & -1 & 0 & a_n(1+\epsilon_{21}) & -a_n\epsilon_{11} & -x_n \\ 0 & 0 & -1 & -\epsilon_{31} & \epsilon_{32} & 0 \\ 0 & 0 & 0 & r_{1_{wx}} & r_{1_{wy}} & 0 \\ 0 & 0 & 0 & r_{2_{wx}} & r_{2_{wy}} & 0 \\ 0 & 0 & 0 & \alpha_{wx} & \alpha_{wy} & -1 \end{bmatrix} \tag{12}$$

where:
$$\begin{cases} \epsilon_{11} &= n_{11} - x_g\epsilon_{31} \\ \epsilon_{12} &= -(n_{20} - x_g\epsilon_{32}) \\ \epsilon_{21} &= n_{02} - y_g\epsilon_{32} \\ \epsilon_{22} &= -(n_{11} - y_g\epsilon_{31}) \\ \epsilon_{31} &= y_g + (y_g\mu_{02} + x_g\mu_{11} + \mu_{21} + \mu_{03})/a \\ \epsilon_{32} &= x_g + (x_g\mu_{20} + y_g\mu_{11} + \mu_{12} + \mu_{30})/a \\ \alpha_{wx} &= (4\mu_{20}\mu_{12} + 2\mu_{20}x_g\mu_{02} + 2\mu_{20}y_g\mu_{11} \\ & \quad -4\mu_{02}\mu_{12} - 2x_g\mu_{02}^2 + 2\mu_{02}y_g\mu_{11} \\ & \quad -4x_g\mu_{11}^2 + 4\mu_{11}\mu_{03} - 4\mu_{11}\mu_{21})/d \\ \alpha_{wy} &= (-4\mu_{20}\mu_{21} - 2y_g\mu_{20}^2 + 2\mu_{11}x_g\mu_{20} \\ & \quad +2\mu_{02}y_g\mu_{20} + 2\mu_{11}x_g\mu_{02} + 4\mu_{02}\mu_{21} \\ & \quad -4\mu_{11}\mu_{12} - 4y_g\mu_{11}^2 + 4\mu_{11}\mu_{30})/d \\ d &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \end{cases}$$

and where $r_{1_{wx}}, r_{1_{wy}}, r_{2_{wx}}$ and $r_{2_{wy}}$ have a too complex analytical form to be given here.

As in the continuous case, and despite the differences in the non constant elements of $\mathbf{L}_{\mathbf{s}}^{\parallel}$, we can note the very nice block triangular form of the obtained interaction matrix. Furthermore, if only the translational degrees of freedom are considered, we have a direct link between each of the first three features and each degree of freedom (note that upper left 3×3 block of $\mathbf{L}_{\mathbf{s}}^{\parallel}$ is equal to $-\mathbf{I}_3$).

Finally, we can use the classical control law:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}_{\mathbf{s}}}^{-1}(\mathbf{s} - \mathbf{s}^*) \tag{13}$$

where $\mathbf{v}$ is the camera velocity sent to the low-level robot controller, λ is a proportional gain, $\mathbf{s}$ is the current value of the visual features computed at each iteration of the control law, $\mathbf{s}^*$ is the desired value of $\mathbf{s}$, and $\widehat{\mathbf{L}_{\mathbf{s}}}$ is chosen as:

$$\widehat{\mathbf{L}_{\mathbf{s}}} = \frac{1}{2}(\mathbf{L}_{\mathbf{s}(\mathbf{s}^*)}^{\parallel} + \mathbf{L}_{\mathbf{s}(\mathbf{s})}^{\parallel})$$

However, control law (13) presents nice decoupling properties only when the desired camera position is such that the image plane is parallel to the object (and for initial camera positions around such configurations). That is why we propose in the next section a new method to generalize this result to the case where the desired camera pose may have any orientation with respect to the object.

IV. GENERALIZATION TO DESIRED OBJECT POSES NON PARALLEL TO THE IMAGE PLANE

The general idea of our method leads to apply a virtual rotation $\mathbf{R}$ to the camera, to compute the visual features after this virtual motion, and to use these transformed features in the control law. More precisely, the rotation is determined so that the image plane of the camera in its desired position is parallel to the object. The decoupling effects obtained when the image plane is parallel to the object can thus be enlarged around any desired configuration of the camera with respect to the object.

A. Image transformation

The first step of our algorithm consists in 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 the selected configuration (but this method necessitates the knowledge of the CAD 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 in an off-line learning step, $\mathbf{R}$ has to be given during this step, in the same way as we set the desired depth Z^* when we consider the parallel case. This method does not necessitate any knowledge on the CAD model of the object. Furthermore, we will see in the experimental results that a coarse approximation of $\mathbf{R}$ is sufficient to obtain nice decoupling properties.

Let us denote (X_t, Y_t, Z_t) and (X, Y, Z) the coordinates of a 3D point after and before the virtual rotational motion. We have of course:

$$\begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} = \mathbf{R}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{14}$$

from which we can immediately deduce the coordinates (x_t, y_t) of the corresponding point in the virtual image that would be 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:

$$\begin{cases} x_t = (r_{11}x + r_{12}y + r_{13})/(r_{31}x + r_{32}y + r_{33}) \\ y_t = (r_{21}x + r_{22}y + r_{23})/(r_{31}x + r_{32}y + r_{33}) \end{cases} \tag{15}$$

where (x, y) are the coordinates of the point in the real image. We can note that transformation (15) can be computed directly from $\mathbf{R}$ and the image point coordinates. An estimation of the coordinates of the 3D point is thus useless.

If the considered object is discrete, transformation (15) is applied for all the N points considered in the desired image (from which visual features denoted $\mathbf{s}_t^*$ are computed), and for all the N points considered in the current image (from which visual features denoted $\mathbf{s}_t$ are computed). Otherwise, that is if we consider a continuous object, the image moments after the virtual rotational motion can be computed by:

$$m_{t_{pq}} = \iint_{\mathcal{D}_t} x_t{}^p y_t{}^q dx_t dy_t$$
$$= \iint_{\mathcal{D}} (\frac{r_{11}x + r_{12}y + r_{13}}{r_{31}x + r_{32}y + r_{33}})^i (\frac{r_{21}x + r_{22}y + r_{23}}{r_{31}x + r_{32}y + r_{33}})^j$$
$$\times det(\mathbf{J_t}) dx dy \qquad (16)$$

$$\text{with: } \mathbf{J_t} = \begin{pmatrix} \frac{\partial x_t}{\partial x} & \frac{\partial x_t}{\partial y} \\ \frac{\partial y_t}{\partial x} & \frac{\partial y_t}{\partial y} \end{pmatrix} \qquad (17)$$

From (16) we can easily prove that the image moments after the virtual rotation are given by:

$$m_{t_{pq}} = \iint_{\mathcal{D}} \frac{(r_{11}x + r_{12}y + r_{13})^i (r_{21}x + r_{22}y + r_{23})^j}{(r_{31}x + r_{32}y + r_{33})^{i+j+3}} dx dy \qquad (18)$$

The control law is finally given by:

$$\mathbf{v} = -\lambda \mathbf{V} \widehat{\mathbf{L_s}}^{-1} (\mathbf{s}_t - \mathbf{s}_t^*) \qquad (19)$$

where matrix $\mathbf{V}$ represents the change of frame of the camera kinematics screw to go from the virtually rotated camera frame to the real one:

$$\mathbf{V} = \begin{bmatrix} \mathbf{R}^T & 0 \\ 0 & \mathbf{R}^T \end{bmatrix}$$

Let us note that model $\widehat{\mathbf{L_s}}$ is now computed with value $\mathbf{s}_t$ and $\mathbf{s}_t^*$:

$$\widehat{\mathbf{L_s}} = \frac{1}{2}(\mathbf{L}_{\mathbf{s}(\mathbf{s_t^*})}^{\parallel} + \mathbf{L}_{\mathbf{s}(\mathbf{s_t})}^{\parallel})$$

which allows to obtain the same decoupling properties as in the parallel case (since matrix $\mathbf{V}$ preserves the decoupling properties between translational and rotational motions).

Finally, we could imagine to estimate the orientation between the camera frame and the object at each iteration of the control law. That would allow to determine a varying transformation such that the decoupling is ensured whatever the camera configuration (and not only around the desired position). That idea will be studied in our future works.

V. EXPERIMENTAL RESULTS

This section presents some experimental results obtained with a six dof eye-in-hand system using first a discrete object and then a continuous object. In the case where the object is discrete, two cases have been considered for the desired camera position: either the image plane is parallel to the object, either it is not. The corresponding images are given on Figure 1.a and 1.b. Furthermore, experiments using a continuous object where the object is non parallel to the image plane (the desired image is given on Fig. 2.a) is also given. The case where a continuous objet is parallel to the image plane has already been presented in [9]. For the parallel case, we have just set approximatively the desired depth Z^* to 0.5 m. For the non parallel case, Z^* has also been set to 0.5 m and rotation $\mathbf{R}$ has been specified by a rotation of 30 dg around x-axis.

A. Pure translational motion

We first consider the case of a pure translational motion. In this experiment, the same translation $T = [-24cm, 17cm, -70cm]$ has been realized for both parallel and non parallel cases (see Figure 1). We have compared the results obtained using the moments proposed in Section 3 as inputs of the control law, and using all the points coordinates (x_k, y_k). In both parallel and non parallel cases, we can see on Figures 1.e, 1.f, 1.g and 1.h the improvements brought using moments and the virtual rotation of the features proposed in Section 4. Indeed, they allow to obtain a pure exponential decrease for the visual features and generate exactly the same camera velocity. As expected, the camera 3D trajectory is thus the same pure straight line in both cases using the proposed features. When points coordinates are used, we have no more a pure exponential decrease for the visual features and for the camera velocity components. The camera trajectory is thus no more a straight line, especially for the non parallel case. Rotational motions are even involved when points are used for the non parallel case.

The results obtained using a continuous object where the object is non parallel to the image plane are given on Figure 2. From those results we also note the same exponential decrease behavior obtained either for the features errors (Fig. 2.c) either for the translational velocities (Fig. 2.d.). Furthermore, the obtained 3D camera trajectory (Fig. 2.e.) is a pure straight line.

B. Complex motion

We now consider a complex motions where both translational and rotational motion are involved. More precisely, the displacement to realize is composed of a translation of approximatively 65 cm and of a rotation of approximatively 30 dg. The initial and desired images are given on Figure 3.a and 1.b (non parallel case and discrete object). For that desired position, and after transformation of the visual features, we have:

$$\mathbf{L}_{\mathbf{s}(\mathbf{s}_t^*)}^{\parallel} = \begin{pmatrix} -1 & 0 & 0 & 0.00 & -0.50 & 0.00 \\ 0 & -1 & 0 & 0.50 & -0.0 & 0.01 \\ 0 & 0 & -1 & -0.00 & -0.01 & 0 \\ 0 & 0 & 0 & 1.49 & -0.56 & 0 \\ 0 & 0 & 0 & 1.58 & 2.50 & 0 \\ 0 & 0 & 0 & 0.11 & -0.17 & -1 \end{pmatrix} \qquad (20)$$

The obtained interaction matrix is sparse. Its condition number is equal to 3.34 while it is 95.61 using points coordinates. The obtained results are given on Figure 3. Despite the large displacement to realize, we can note the good behavior of the features errors mean and of the camera velocities: they decrease exponentially with no oscillation (see Figure 3.b and Figure 3.d). The 3D camera trajectory is also satisfactory (see Figure 3.c).

The experimental results using a continuous objet are given on Figure 4. An exponential decrease of the velocities and features errors mean is also obtained. The 3D camera trajec-
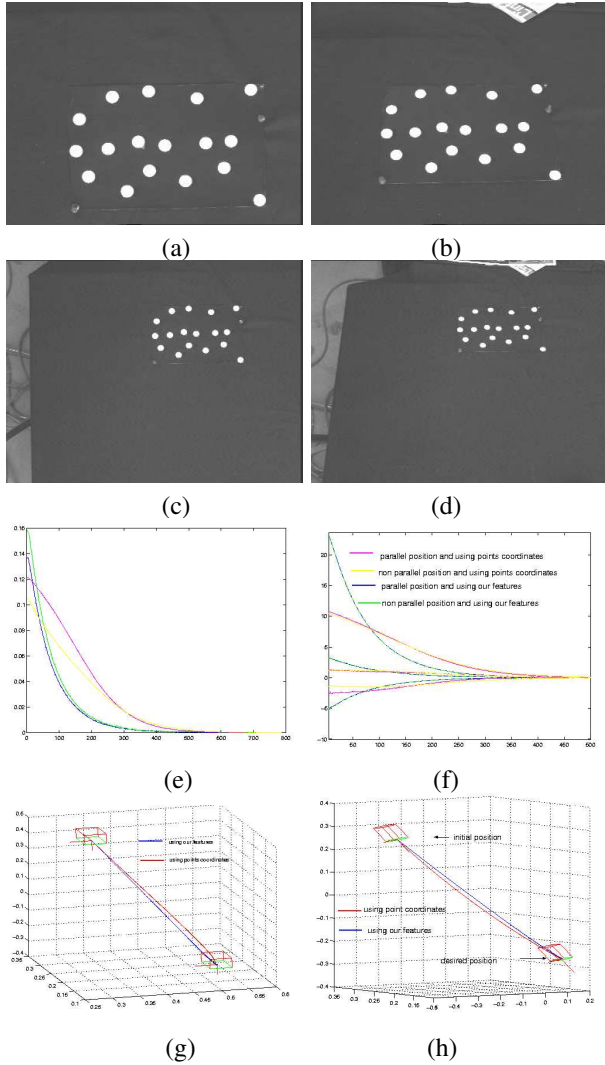
(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 1.   (a) desired image where the object is parallel to the image plane), (b) desired image where the object is not parallel to the image plane), (c) initial image for a pure translational motion between (a) and (c)), (d) initial image for a pure translational motion between (b) and (d)), (e) features errors mean, (f) velocities, (g) camera 3D trajectory (object parallel to the image plane) (h) camera 3D trajectory (object non parallel to the image plane).

tory is adequate despite of the large displacement between the desired and the initial position of the camera.

C. Results with a bad camera 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 for the desired camera position ($\widehat{Z}^* = 0.8\ m$ instead of $Z^* = 0.5\ m$). Furthermore, an error equal to 10 dg has been set in $\mathbf{R}$. The obtained results are given on Figure 5. The behavior of the system is very similar to that of the previous experiment, which validates the robustness of our scheme with respect to such errors.
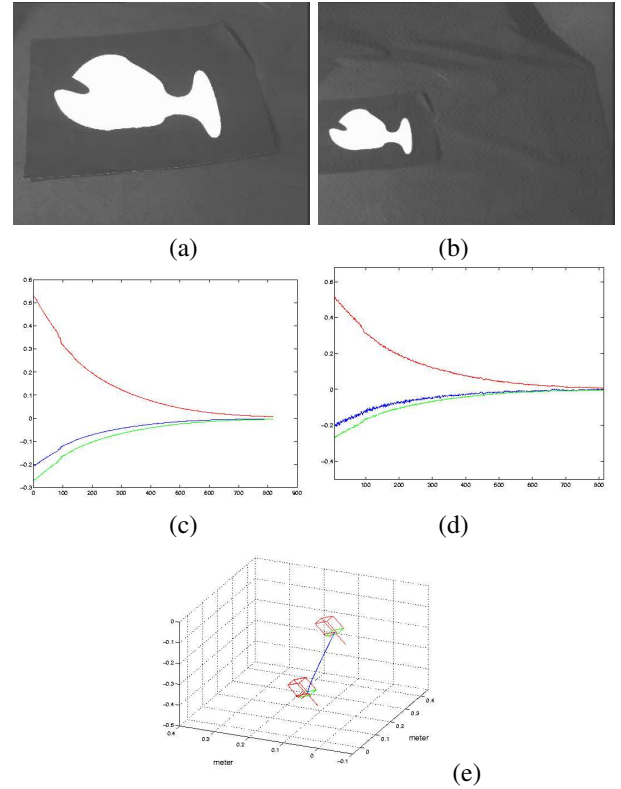

(a)

(b)

(c)

(d)

(e)

Fig. 2.   Results for a pure translation using a continuous object (when the object is not parallel to the image plane): (a) desired image, (b) initial image, (c) features errors, (d) translational velocities, (e) camera 3D trajectory
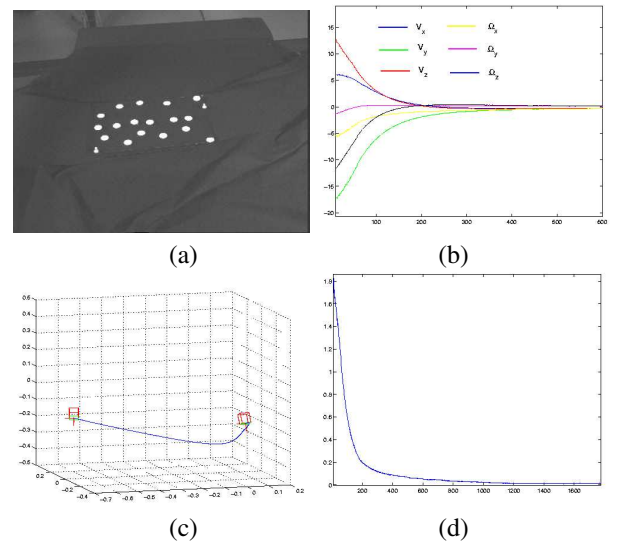

(a)

(b)

(c)

(d)

Fig. 3.   Results for a complex motion and using a discrete object: (a) initial image, (b) velocities, (c) camera trajectory, (d) features errors mean

D. Results for complex images

In this paragraph, we present similar experimental results obtained on more complex images (see Figure 6). The considered points have been extracted using Harris detector and tracked using a SSD algorithm. We can note however that

(a)   (b)

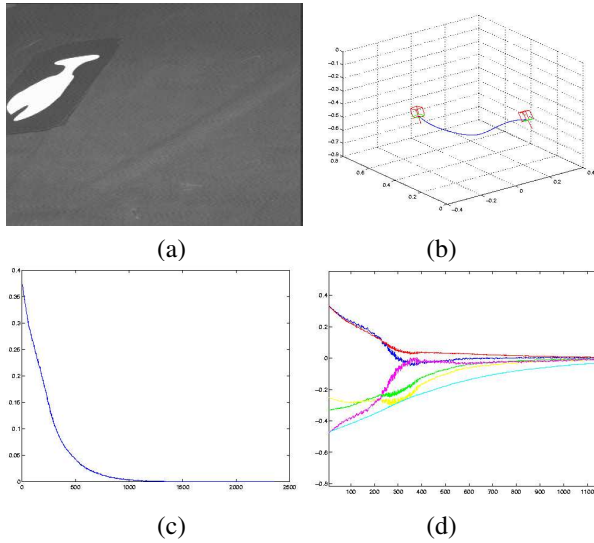(c)   (d)

Fig. 4. Results for a complex motion using a continuous object: (a) initial image, (b) camera 3D trajectory, (c) features errors mean, (d) camera velocities.

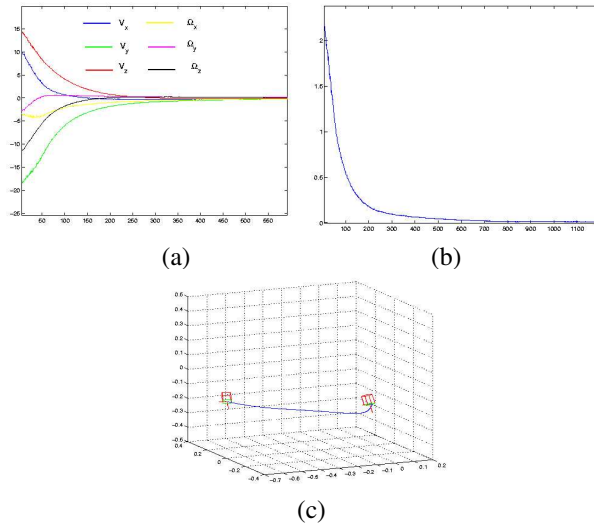(a)   (b)

(c)

Fig. 5. Results with modeling errors: (a) velocities, (b) features errors mean, (c) camera trajectory.

the plots are more noisy than using simple dots, because of the less accurate points extraction. It is mainly noticeable on the ω_x and ω_y components of the camera velocity whose value depends on 5th order moments (while ω_z and υ_z are not noisy at all since their value only depend of 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 can be improved using a sub pixel accuracy image tracker such as Shi-Tomasi algorithm [8].

VI. CONCLUSION

In this paper, we have proposed a new visual servoing scheme based on image moments for objects composed of a set of discrete points. Six features have been designed to

(a)   (b)
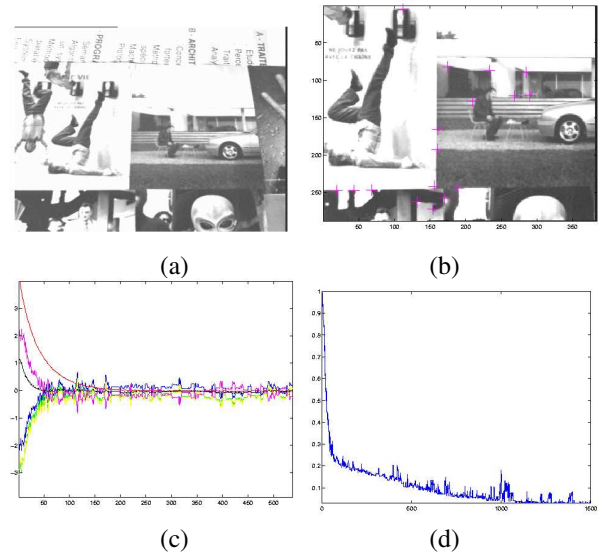
(c)   (d)

Fig. 6. Results for complex images: (a) initial image, (b) desired image, (c) velocities, (d) features errors mean

decouple the camera degrees of freedom, which allows one to obtain a large domain of convergence, a good behavior of the visual features in the image, as well as 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 considered object. This method is valid either for discrete objects either for continuous objects. The experimental results show the validity and the robustness of our approach with respect to modeling errors. Furthermore, in the case of a discrete objet, our method requires only that the set of points considered in the desired image is the same in the initial and tracked images (no matching between these points is necessary to compute the moments).

REFERENCES

[1] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In A.S. Morse D. Kriegman, G. Hager, editor, *The Confluence of Vision and Control*, number 237, pages 66–78. Springer-Verlag, 1998. LNCIS.
[2] P. I. Corke and S. A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Trans. on Robotics and Automation*, 17(4):507–515, Aug. 2001.
[3] G. Hager and K. Toyoma. The xvision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23–37, 1997.
[4] C. Harris and M. Stephens. A combined corner and edge detector. In *In proceeding of the 4th Alvey Vision Conference*, pages 147–151, 1988.
[5] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, Oct. 1996.
[6] N. Okiyama M. Iwatsuki. A new formulation of visual servoing based on cylindrical coordinates system with shiftable origin. In *IEEE/RSJ IROS*, Lausanne, Switzerland, Oct. 2002.
[7] R. Mahony, P. Corke, and F. Chaumette. Choice of image features for depth-axis control in image-based visual servo control. In *IEEE/RSJ IROS'02*, pages 390–395, Lausanne, Switzerland, Oct. 2002.
[8] J. Shi and C. Tomasi. Good features to track. In *CVPR'94*, pages 593–600, Seattle, June 1994.
[9] O. Tahri and F. Chaumette. Application of moment invariants to visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'03*, pages 4276–4281, Taipeh, Sep. 2003.