HAL
archives-ouvertes.fr

New decoupled visual servoing scheme based on invariants projection onto a sphere

O. Tahri, François Chaumette, Youcef Mezouar

To cite this version:

O. Tahri, François Chaumette, Youcef Mezouar. New decoupled visual servoing scheme based on invariants projection onto a sphere. IEEE Int. Conf. on Robotics and Automation, ICRA'08, 2008, Pasadena, California, France. pp.3238-3243, 2008. <inria-00351876>

HAL Id: inria-00351876

https://hal.inria.fr/inria-00351876

Submitted on 12 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

New Decoupled Visual Servoing Scheme based on Invariants from Projection onto a Sphere

Omar Tahri, François Chaumette and Youcef Mezouar

***Abstract*— In this paper a new decoupled imaged-based control scheme is proposed from projection onto a unit sphere. This control scheme is based on moment invariants to 3D rotational motion. This allows the control of translational motion independently of the rotational one. First, the analytical form of the interaction matrix related to the spherical moments is derived. It is based on the projection of a set of points onto a unit sphere. From the spherical moment, six features are presented to control the full 6 degrees of freedom. Finally, the results are validated through realistic simulation results.**

I. INTRODUCTION

In image-based visual servoing, the choice of the set of visual features to be used in the control scheme is still an open question, despite of the large quantity of results obtained in the last few years. Indeed, one would like to choose features which provide optimal system behavior. Whatever the nature of the possible measures extracted from the image, wether it be a set of image points coordinates or 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 of different features 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 task singularities [2].

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 analyzed theoretically. 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. To overcome these problems, it is possible to combine path planning and visual servoing, since tracking planned trajectories allows the error to always remain small [12], [15]. A second approach is to use the measures to build particular visual features that will ensure expected properties of the control scheme. Several works have been realized in image-based visual servoing following the same general objective. In [14], a vanishing point and the horizon line have been selected. This choice ensures a good decoupling between translational and rotational dofs. In [9], vanishing points have also been used for a dedicated object (a 3D rectangle), once again for decoupling properties. For the same object, six visual features have been designed in [4] to control the six dofs of a robot arm, following a partitioned approach. In [8], the coordinates of points are expressed in a cylindrical coordinate system instead of the classical Cartesian one, so as to improve the robot trajectory. In [7], 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 under-actuated system. Recently, [18] proposed a decoupled visual servoing from spheres using a spherical projection model.

In previous works, the analytical form of the interaction matrix related to any image moment with a conventional camera and corresponding to planar objects has been computed. This makes possible to consider planar objects of any shape [3], [17]. If a collection of points is measured in the image, moments can also be used [17]. In both cases, moments allow the use of intuitive geometrical features, such as the center of gravity or the orientation of an object. By selecting an adequate combination of moments, it is then possible to determine partitioned systems with good decoupling and linearizing properties [3], [17]. For instance, using such features, the interaction matrix block corresponding to the translational velocity can be a constant block diagonal. The use of moments instead of point coordinates may solve several problems, such as, for instance, local minima. This has significantly improved the 3D behavior of the system. However, the previously published works only concerned planar objects and conventional perspective cameras. In this paper, the proposed decoupled control scheme is valid for sets of non coplanar points. Furthermore, it is also valid for any sensor obeying the unified camera model. In other words, it encompasses all sensors in this class [11], [6]: perspective and catadioptric. Some class of fisheye cameras can also be concerned by this model [5], [11].

Image moments have been widely studied in the computer vision community, especially for pattern recognition applications. Indeed, invariance to some transformations such as scale, 2D translation and/or 2D rotation can be obtained by adequate combinations of moments. This property is of great interest in pattern recognition. This explains the amount of work invested in deriving moment invariants (see [16], [10] for instance). Such invariance property is also of particular interest in visual servoing. In previous work [17], 2D moment invariants served to obtain such decoupled control schemes. More precisely, these works were mainly based on invariants to translation and rotation around the optical axis. In this

O. Tahri and Y. Mezouar are with LASMEA, Université Blaise Pascal, 63177 AUBIERE, France. firstname.lastname@lasmea.univ-bpclermont.fr
F. Chaumette is with IRISA/INRIA, campus de Beaulieu, 35042 Rennes, France. firstname.lastname@irisa.fr

work, the 3D moments computed from the projection onto the unit sphere are used to develop a new decoupled control scheme.

In the next section, the unified camera model is recalled. In Section III, the analytical form of the interaction matrix related to moments on the sphere is determined. Furthermore, a new vector of six features to control the six camera degrees of freedom is proposed. Finally, in Section V, several simulation results are presented to validate our approach.

II. Central catadioptric camera model

In this section a slightly modified version of the projection model of Geyer [6] and Barreto [1] is recalled [11]. The projection of 3D points can be done in the following steps (see Figure 1):

1) world points in the mirror frame are projected onto the unit sphere,

$$(\boldsymbol{\mathcal{X}}_{\mathcal{F}_m}) \rightarrow (\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_m} = \frac{\boldsymbol{\mathcal{X}}}{\|\boldsymbol{\mathcal{X}}\|} = (x_s,\ y_s,\ z_s) \qquad (1)$$

2) the points coordinates are then changed to a new reference frame centered in $\mathbf{p} = (0,\ 0,\ -\xi)$:

$$(\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_m} \rightarrow (\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_p} = (x_s,\ y_s,\ z_s + \xi) \qquad (2)$$

3) the point is then projected onto the normalized plane,

$$\mathbf{m} = (x,\ y,\ 1) = \left(\frac{x_s}{z_s+\xi},\ \frac{y_s}{z_s+\xi},\ 1\right) = \hbar(\boldsymbol{\mathcal{X}}_s) \qquad (3)$$

where the function $\hbar$ is one-to-one and such that

$$\hbar^{-1}(\mathbf{m}) = \begin{bmatrix} \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{bmatrix} \qquad (4)$$

4) the final projection involves a generalized camera projection matrix $\mathbf{K}$ (with f the focal length, (u_0, v_0) the principal point, s the skew and r the aspect ratio)

$$\mathbf{p} = \mathbf{K}\mathbf{m} = \begin{bmatrix} \gamma & \gamma s & u_0 \\ 0 & \gamma r & v_0 \\ 0 & 0 & 1 \end{bmatrix} = k(\mathbf{m}) \qquad (5)$$

	ξ	γ
Parabola	1	$-2pf$
Hyperbola	$\frac{df}{\sqrt{d^2+4p^2}}$	$\frac{-2pf}{\sqrt{d^2+4p^2}}$
Ellipse	$\frac{df}{\sqrt{d^2+4p^2}}$	$\frac{2pf}{\sqrt{d^2+4p^2}}$
Planar	0	-f
Perspective	0	f
d: distance between focal points $4p$: latus rectum		

TABLE I
Unified model parameters

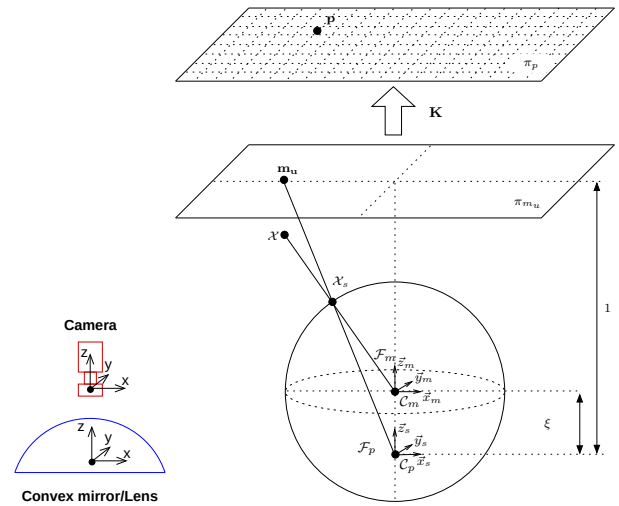

Fig. 1. Unified image formation (on the right), axis convention (on the left)

The different possible values for ξ and γ are recalled on Table I. Note that the conventional perspective camera is nothing but a particular case of this model (when $\xi = 0$). In this paper, the whole calibration parameters are supposed to be known using a sensor calibration method ([11] for instance). In this case, the projection onto the unit sphere from the image plane is possible for all sensors obeying this model. Furthermore, geometrically correct perspective images from the pictures captured by an omnidirectional camera can be generated.

III. Moments from projection onto a sphere

In this section, moments on the unit sphere are first defined and their interaction matrix is determined.

A. 3D Moment definition

The 3D moment computed from a discrete set of points are defined by the following classical equation:

$$m_{s_{i,j,k}} = \sum_{h=1}^{N} x_{s_h}^i\ y_{s_h}^j\ z_{s_h}^k \qquad (6)$$

where $(x_s,\ y_s,\ z_s)$ are the coordinates of a 3D point. In our application, these coordinates are nothing but the coordinates of a point projected onto the unit sphere. They can be computed from the projection of a point onto the image plane and the inverse transform (4). In order to derive the interaction matrix of $m_{s_{i,j,k}}$, the interaction related to the point coordinates of the point projection onto the unit sphere is recalled in the next paragraph.

B. Interaction matrix of point on the sphere

It is well known that the interaction matrix $\mathbf{L}_{\mathbf{x_s}}$ of point $\mathbf{x_s}$ on the unit sphere (defined such that $\dot{\mathbf{x}}_\mathbf{s} = \mathbf{L}_{\mathbf{x_s}}\mathbf{V}$ where $\mathbf{V} = (\mathbf{v}, \boldsymbol{\omega})$ is the sensor instantaneous velocity) is given by [7], [16], [18]:

$$\mathbf{L_{X_s}} = \begin{bmatrix} -\frac{1}{r}+\frac{x_s^2}{r^2} & \frac{x_s y_s}{r^2} & \frac{x_s z_s}{r^2} & 0 & -z_s & y_s \\ \frac{x_s y_s}{r^2} & -\frac{1}{r}+\frac{y_s^2}{r^2} & \frac{y_s z_s}{r^2} & z_s & 0 & -x_s \\ \frac{x_s z_s}{r^2} & \frac{y_s z_s}{r^2} & -\frac{1}{r}+\frac{z_s^2}{r^2} & -y_s & x_s & 0 \end{bmatrix} \tag{7}$$

where $r = \sqrt{X^2+Y^2+Z^2}$, (X, Y, Z) being the coordinates of the point that has been projected on the unit sphere. From the block corresponding to rotational motion, it can be seen that the projection of a point onto a sphere behaves as a 3D point with respect to rotational motion. Indeed, a rotational motion in 3D produces the same rotational motion of the projection onto the sphere. This means that the moment invariants to rotation in 3D space are also invariant if the considered points are projected onto the unit sphere. This important property will be used to select three features invariant to rotation in order to control the three translational degrees of freedom. For instance, the following combinations are invariant to rotational motions [16]:

$$I_1 = m_{200}m_{020} - m_{200}m_{002} + m_{110}^2 + m_{101}^2 - m_{020}m_{002} + m_{011}^2 \tag{8}$$

$$\begin{aligned} I_2 = &- m_{300}m_{120} - m_{300}m_{102} + m_{210}^2 - m_{210}m_{030} - m_{210}m_{012} \\ &+ m_{201}^2 - m_{201}m_{021} - m_{201}m_{003} + m_{120}^2 - m_{120}m_{102} \\ &+ 3m_{111}^2 + m_{102}^2 - m_{030}m_{012} + m_{021}^2 - m_{021}m_{003} + m_{012}^2 \end{aligned} \tag{9}$$

$$\begin{aligned} I_3 = & m_{300}^2 + 3m_{300}m_{120} + 3m_{300}m_{102} + 3m_{210}m_{030} \\ &+ 3m_{210}m_{012} + 3m_{201}m_{021} + 3m_{201}m_{003} + 3m_{120}m_{102} \\ &- 3m_{111}^2 + m_{030}^2 + 3m_{030}m_{012} + 3m_{021}m_{003} + m_{003}^2 \end{aligned} \tag{10}$$

In the next section, the interaction matrix of moments defined from the projection of points onto the unit sphere is derived.

C. Interaction matrix of moment onto the unit sphere

The derivative of (6) with respect to time is given by:

$$\begin{aligned} \dot{m}_{s_{i,j,k}} = \sum_{h=0}^{N} (& i\ x_{s_h}^{i-1}\ y_{s_h}^{j}\ z_{s_h}^{k}\dot{x}_{s_h} + j\ x_{s_h}^{i}\ y_{s_h}^{j-1}\ z_{s_h}^{k}\dot{y}_{s_h} \\ &+ k\ x_{s_h}^{i}\ y_{s_h}^{j}\ z_{s_h}^{k-1}\dot{z}_{s_h}) \end{aligned} \tag{11}$$

Let us assume now that a set of coplanar points is concerned. The point coordinates belonging to this plane holds the following equation:

$$AX + BY + CZ = 1 \tag{12}$$

By dividing each side of (12) by $r = \sqrt{X^2+Y^2+Z^2}$, the plane equation can be rewritten using the point projection coordinates onto the unit sphere as follow:

$$\frac{1}{r} = Ax_s + By_s + Cz_s \tag{13}$$

In fact, this assumption will not limit the application of the following results to planar objects only. Indeed, a non coplanar set of points can be divided to several subsets of three non-collinear points (or more than three if available) to define several planes. The moments derived from the projection onto unit sphere will be computed for each subset of points. By combining (7) and (13) with (11) and after tedious calculus, the interaction matrix related to $m_{s_{i,j,k}}$ can be determined [16]:

$$\mathbf{L}_{m_{s_{i,j,k}}} = \begin{bmatrix} m_{s_{vx}} & m_{s_{vy}} & m_{s_{vz}} & m_{s_{wx}} & m_{s_{wy}} & m_{s_{wz}} \end{bmatrix} \tag{14}$$

where:

$$\begin{cases} m_{s_{vx}} = & A(\beta_d m_{s_{i+2,j,k}} - i m_{s_{i,j,k}}) \\ & + B(\beta_d m_{s_{i+1,j+1,k}} - i m_{s_{i-1,j+1,k}}) \\ & + C(\beta_d m_{s_{i+1,j,k+1}} - i m_{s_{i-1,j,k+1}}) \\ m_{s_{vy}} = & A(\beta_d m_{s_{i+1,j+1,k}} - j m_{s_{i+1,j-1,k}}) \\ & + B(\beta_d m_{s_{i,j+2,k}} - j m_{s_{i,j,k}}) \\ & + C(\beta_d m_{s_{i,j+1,k+1}} - j m_{s_{i,j-1,k+1}}) \\ m_{s_{vz}} = & A(\beta_d m_{s_{i+1,j,k+1}} - k m_{s_{i+1,j,k-1}}) \\ & + B(\beta_d m_{s_{i,j+1,k+1}} - k m_{s_{i,j+1,k-1}}) \\ & + C(\beta_d m_{s_{i,j,k+2}} - k m_{s_{i,j,k}}) \\ m_{s_{wx}} = & j m_{s_{i,j-1,k+1}} - k m_{s_{i,j+1,k-1}} \\ m_{s_{wy}} = & k m_{s_{i+1,j,k-1}} - i m_{s_{i-1,j,k+1}} \\ m_{s_{wz}} = & i m_{s_{i-1,j+1,k}} - j m_{s_{i+1,j-1,k}} \end{cases}$$

with $\beta_d = i + j + k$. From this interaction matrix it can be shown that the interaction matrices related to the invariants given by (8), (9) and (10) have the following form:

$$\mathbf{L}_I = \begin{bmatrix} i_{v_x} & i_{v_y} & i_{v_z} & 0 & 0 & 0 \end{bmatrix} \tag{15}$$

This means that these features only depend on the translational motion. In the next section, six features are derived from the image moments to control the full six dofs.

IV. Features choice

The main objective of visual features selection is to obtain a sparse 6×6 full rank interaction matrix. In the next paragraph, three features to control the rotational motions are firstly presented.

A. Features to control rotational motions

To control the rotational degrees of freedom we could think to use the center of gravity of the object projection onto the unit sphere:

$$\mathbf{x_{s_g}} = \begin{pmatrix} x_{s_g}, & y_{s_g}, & z_{s_g} \end{pmatrix} = \begin{pmatrix} \frac{m_{100}}{m_{000}}, & \frac{m_{010}}{m_{000}}, & \frac{m_{001}}{m_{000}} \end{pmatrix}$$

From (14), the interaction matrices related to those coordinates can be deduced. We obtain:

$$\mathbf{L_{xs_g}} = \begin{bmatrix} A(\eta_{300} - x_{s_g}) + B(\eta_{210} - y_{s_g}) + C(\eta_{201} - z_{s_g}) \\ A\eta_{210} + B\eta_{120} + C\eta_{111} \\ A\eta_{201} + B\eta_{111} + C\eta_{102} \\ 0 \\ -z_{s_g} \\ y_{s_g} \end{bmatrix}^{\top} \tag{16}$$

$$\mathbf{L_{ys_g}} = \begin{bmatrix} A\eta_{210} + B\eta_{120} + C\eta_{111} \\ A(\eta_{120} - x_{s_g}) + B(\eta_{030} - y_{s_g}) + C(\eta_{021} - z_{s_g}) \\ A\eta_{111} + B\eta_{021} + C\eta_{012} \\ z_{s_g} \\ 0 \\ -x_{s_g} \end{bmatrix}^\top \tag{17}$$

$$\mathbf{L_{zs_g}} = \begin{bmatrix} A\eta_{201} + B\eta_{111} + C\eta_{102} \\ A\eta_{111} + B\eta_{021} + C\eta_{012} \\ A(\eta_{102} - x_{s_g}) + B(\eta_{012} - y_{s_g}) + C(\eta_{003} - z_{s_g}) \\ -y_{s_g} \\ x_{s_g} \\ 0 \end{bmatrix}^\top \tag{18}$$

where $\eta_{ijk} = \frac{m_{ijk}}{m_{000}}$. In fact, only two coordinates of $\mathbf{x_{s_g}}$ are useful for the control. Indeed, since $\mathbf{x_{s_g}}$ belongs to the unit sphere, its coordinates hold the sphere equation constraint $x_{s_g}^2 + y_{s_g}^2 + z_{s_g}^2 = 1$, which makes only two coordinates to be independent. We have chosen to use the two first coordinates x_{s_g} and y_{s_g} to mainly control the rotational motion around the camera x-axis and y-axis. To control the rotational motion around the optical axis, the orientation of the object projection $\alpha_z = \frac{1}{2}\arctan(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}})$ is used as already proposed in [4], [17], [3]. Since the camera is calibrated, $\alpha_z = \frac{1}{2}\arctan(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}})$ can be computed using (4). The interaction matrix corresponding to α_z for a planar object in parallel position with respect to the image plane is given by:

$$\mathbf{L}^{\parallel}_{\alpha_z s} = \begin{bmatrix} 0 & 0 & 0 & \alpha_{wx} & \alpha_{wy} & -1 \end{bmatrix}$$

Note that α_z depends mainly on rotation around the optical axis. To summarize, the following features vector is used to control the rotational motions:

$$\mathbf{s}_\omega = \begin{bmatrix} x_{s_g} \\ y_{s_g} \\ \alpha_z \end{bmatrix} \tag{19}$$

In the next paragraph, the choice of three features to control the translational motion is detailed.

B. Features to control translational motions

As mentioned previously, the invariants to 3D rotation such those given by (8), (9) and (10) will be considered to select three features in order to control the translational dofs. The set of points is firstly organized into subsets containing at least 3 points defining a plane. Note that this can be performed even with only 4 available points, since it is possible to define 4 different combinations of 3 points from only 4 non coplanar points. An example using only 4 points will be presented in the next section.

The invariants to rotational motion (8), (9) and (10) can be computed for each subset of points as well as their related interaction matrices from (14). A first possible combination of features to control the translational motion can be obtained by stacking the features computed from (8), (9) and (10) for each subset of points. In this case, more than three features will be available. More precisely, the number of available features is three time the number of subset of points. This could cause some local minima problem. Furthermore, the invariants (9) and (10) are of higher orders than (8). Therefore, they are more sensitive to noise [13], [19]. In fact, to select three independent features in order to control the translation, we have chosen to use (8) for each subset. At this level, the number of features is equal to the number of points subsets. To obtain exactly three independent features, the subsets of point can be compounded onto three main subsets, such that their gravity centers are non-collinear. In the next section, validation results will be presented.

V. Simulation results

In the whole simulation presented in the following, the current value of the interaction matrix will be used in the control law:

$$\mathbf{V} = -\lambda \mathbf{L_s}^{+}(\mathbf{s} - \mathbf{s}^*) \tag{20}$$

The plane parameters corresponding to each subset of points can be estimated using (13) and the approximations of the point depths to the sphere center r. They will be assumed known in the following simulations.

In the following, the camera position is controlled using a set of 9 non coplanar points. The rotational and the translational motions between the initial and the desired positions of the camera are given respectively by (21) and (22).

$$\boldsymbol{\theta}\mathbf{u} = \begin{bmatrix} 17.65 & 0 & 88.25 \end{bmatrix}^o \tag{21}$$

$$\mathbf{t_1} = \begin{bmatrix} -0. & -0. & -0. \end{bmatrix} \tag{22}$$

Recall that the rotation matrix is obtained from the rotation vector $\theta\mathbf{u}$ using the well known Rodrigues' formula. In this simulation, a comparison of the behavior of the system using points coordinates or moments is performed. In fact, the presented results using moments concern conventional cameras as well as omnidirectional ones, since the visual servoing will be performed onto the unit sphere. In this case, only the behavior of the points in image will depend on the kind of the camera. The plots of the obtained results are given on Figure 2. From Figures 2.a and 2.b, it can be observed that a nice decrease of the feature errors is obtained using both moment or points coordinates in the control law. On the contrary, a better behavior of the velocities is obtained using moments as features (see Figs. 2.c, 2.d, 2.e, 2.f). Indeed, in the case where the points coordinates are used as features, it can be noticed that the velocities suffer from oscillations. Furthermore, since the considered translational motion is null, the translational velocity computed using the moments is also null (thanks to the invariants to 3D rotations). On the contrary, the translational velocity computed using the points coordinates as features is not null. Indeed, it can be seen from 2.c that the computed translational velocity is very strong, especially with respect to the optical axis of the camera (nearly $20cm/s$ for the first iteration).

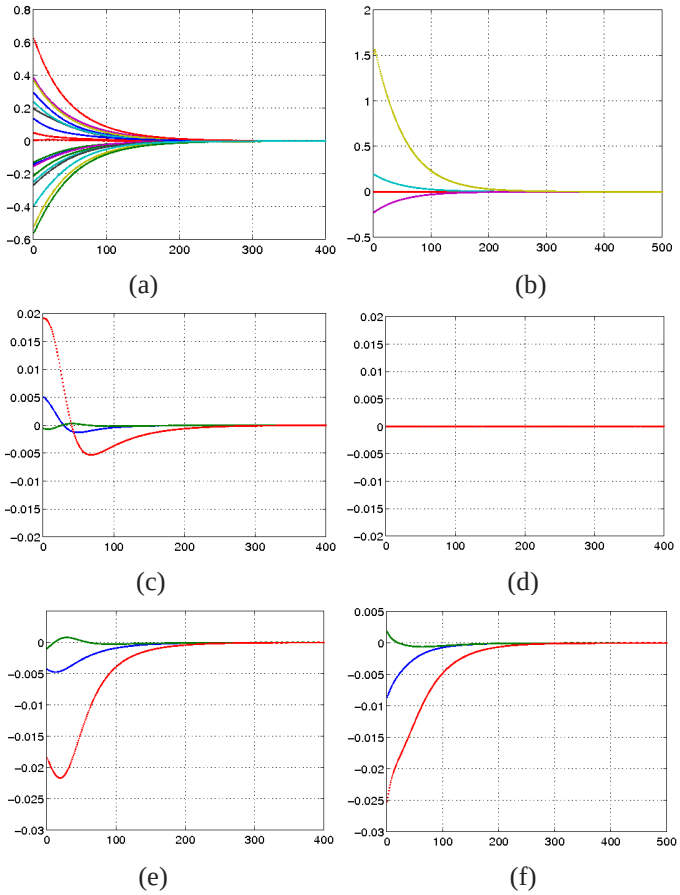

Fig. 2. Comparison results: a) features errors using point coordinates as features, b) feature errors using moments as features, c) translational velocities (m/s) using points coordinates as features, d) translational velocities (m/s) using moments as features, e) rotational velocities (rad/s) using point coordinates as features, f) rotational velocities (rad/s) using moments as features

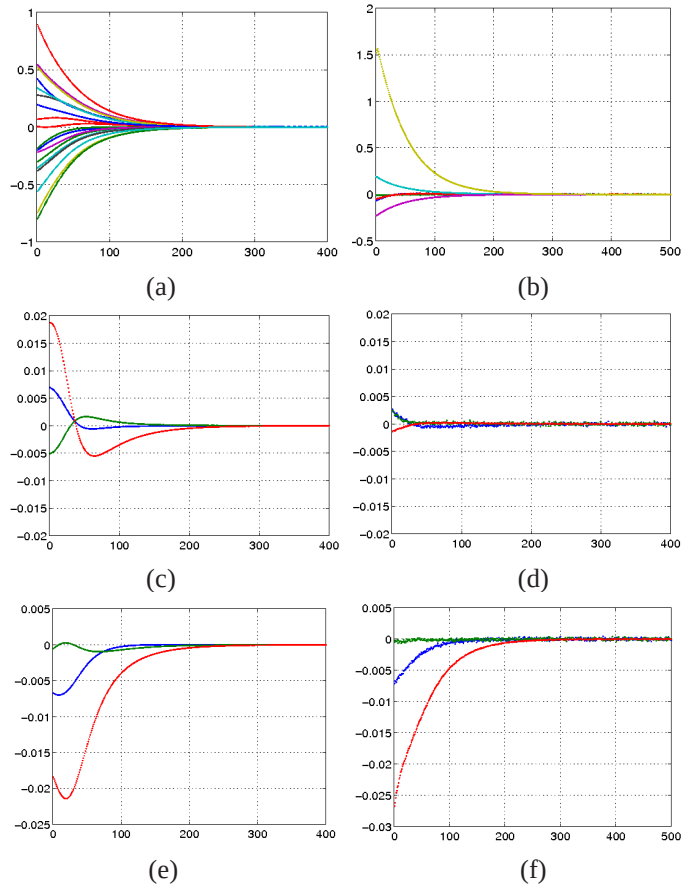

Fig. 3. Results with bad camera calibration, corrupted 3D data and image noise: a) features errors using point coordinates as features, b) feature errors using moments as features, c) translational velocities (m/s) using points coordinates as features, d) translational velocities (m/s) using moments as features, e) rotational velocities (rad/s) using point coordinates as features, f) rotational velocities (rad/s) using moments as features

We now test the robustness of our approach with respect to camera and plane parameters errors. The considered positioning task correspond once again to a generic motion described by (21) and (22). In this simulation, errors have been added to camera intrinsic parameters. More precisely, 30% on the focal length and 25 pixels on the coordinates of the principal point of a conventional perspective camera. Furthermore, a random noise with variance of $1pixel$ is added to each point in the image and the plane parameters have been corrupted with an error up to 20% of their amplitudes.

Figure 3 shows the obtained results using both moments or points coordinates as features. From this figure, it can be seen that a satisfactory behavior is obtained using the moments as features, despite of errors on camera calibration, plane parameters and the image noise. Indeed, even if not null, the computed translational velocities using the invariants to 3D rotation are still closer to zero (see Figure 3.d). Furthermore, nice behavior is still obtained for the feature errors as well as for the rotational velocities (see Figures 3.b and 3.f). On the contrary, a strong more oscillations are observed on translational and rotational velocities using points coordinates as features (see Figures 3.c and 3.e).

In the last simulation, a set of only 4 non coplanar points is considered. Three subsets of points are defined from three different combinations of the 3 points within the 4 initial points. The rotational and translational motions to perform are given by (23) and (24)

$$\boldsymbol{\theta}\mathbf{u}_3 = \left[\begin{array}{ccc} -19.40 & 19.40 & -29.10 \end{array}\right]^o \tag{23}$$

$$\mathbf{t_2} = \left[\begin{array}{ccc} -0.4 & -0.4 & -0.2 \end{array}\right] meter \tag{24}$$

Figure 4.a and 4.b shows that the system converges to its desired position using both points coordinates (case of conventional camera) or moments as features. Furthermore, from Figures 4.c and 4.e the velocities computed using the points coordinates as features suffer oscillations. On the contrary, a nice decrease is obtained using moments (see Figures 4.d and 4.f)

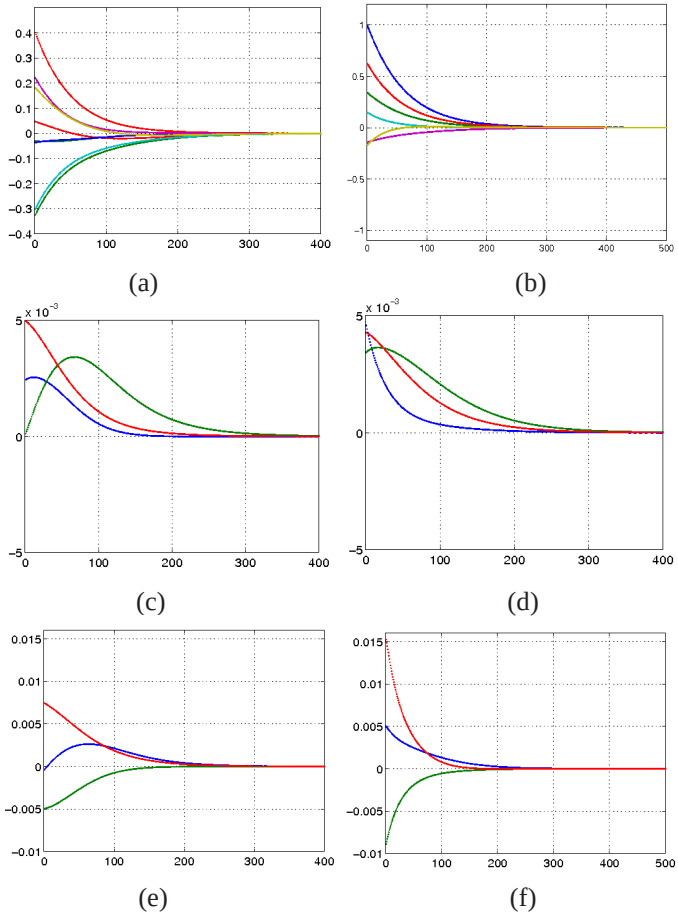

Fig. 4. Comparison results using 4 coplanar points: a) features errors using point coordinates as features, b) feature errors using moments as features, c) translational velocities (m/s) using points coordinates as features, d) translational velocities (m/s) using moments as features, e) rotational velocities (rad/s) using point coordinates as features, f) rotational velocities (rad/s) using moments as features

VI. Conclusions and Future Works

In this paper, a new decoupled image-based control using the projection onto unit sphere was proposed. The 3D moment invariants to rotation are employed to control independently the translations. On the contrary of the previous published works, this decoupled control scheme is valid for a coplanar set of points as well as non coplanar ones. It is also valid for all cameras obeying the unified camera model. Futures works will be devoted to extend this scheme to the continuous case where the object is defined by regions in image instead of a set of points. They will also be concerned with the invariants selection such that the interaction block corresponding to the translational motions is diagonal.

References

[1] J. P. Barreto and H. Araujo. Issues on the geometry of central catadioptric image formation. In *Computer Vision and Pattern Recognition*, volume 2, pages 422–427, Hawaii, USA, December 2001.

[2] F. Chaumette. Potential problems of stability and convergence in imagebased and position-based visual servoing. In Springer-Verlag, editor, *The Confluence of Vision and Control*, volume 237 of *LNCIS*, pages 66–78, 1998.

[3] F. Chaumette. Image moments: A general and useful set of features for visual servoing. *IEEE Transaction on Robotics and Automation*, 20(4):713723, August 2004.

[4] P. I. Corke and S. A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transaction on Robotics and Automation*, 17(4):507–515, August 2001.

[5] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A generic fisheye camera model for robotic applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, California, USA, 29 October 2 November 2007.

[6] C. Geyer and K. Daniilidis. Mirrors in motion: Epipolar geometry and motion estimation. *International Journal on Computer Vision*, 45(3):766–773, 2003.

[7] T. Hamel and R. Mahony. Visual servoing of an under-actuated dynamic rigid body system: an image-based approach. *IEEE Transaction on Robotics and Automation*, 18(2):187–198, April 2002.

[8] M. Iwatsuki and N. Okiyama. A new formulation of visual servoing based on cylindrical coordinates system with shiftable origin. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, October 2002.

[9] J.-S. Lee, I. Suh, B.-J. You, and S.-R. Oh. A novel visual servoing approach involving disturbance observer. In *IEEE International Conference on Robotics and Automation, ICRA99*, pages 269–274, Detroit, Michigan, May 1999.

[10] A. G. Mamistvalov. n-dimensional moment invariants and conceptual mathematical theory of recognition n-dimensional solids. *IEEE Transaction Pattern Analysis and Machine Intelligence*, 20(8):819–831, 1998.

[11] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, Roma, Italy, April 2007.

[12] Y. Mezouar and F. Chaumette. Path planning for robust image-based control. *IEEE Transaction on Robotics and Automation*, 18(4):534–549, August 2002.

[13] R. Mukundan and K. R. Ramakrishnan. *oment Functions in Image Analysis Theory and Application*. Co.Pte.Ltd, 1998.

[14] P. Rives and J. Azinheira. Linear structures following by an airship using vanishing points and horizon line in a visual servoing scheme. In *IEEE International Conference on Robotics and Automation, ICRA04*, pages 255–260, New Orleans, Louisiana, April 2004.

[15] F. Schramm, F. Geffard, G. Morel, and A. Micaelli. Calibration free image point path planning simultaneously ensuring visibility and controlling camera path. In *IEEE International Conference on Robotics and Automation*, pages 2074–2079, Roma, April 2007.

[16] O. Tahri. *Utilisation des moments en asservissement visuel et en calcul de pose*. PhD thesis, University of Rennes, 2004.

[17] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transaction on Robotics*, 21(6):1116–1127, December 2005.

[18] R. Tatsambon Fomena and F. Chaumette. Visual servoing from spheres using a spherical projection model. In *IEEE International Conference on Robotics and Automation, ICRA'07*, Roma, Italia, April 2007.

[19] C. Teh and R. T. Chin. On image analysis by the method of moments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.