Generic Decoupled Image-Based Visual Servoing for Cameras Obeying the Unified Projection Model

Omar Tahri, Youcef Mezouar, François Chaumette and Peter Corke

Abstract— In this paper a generic decoupled imaged-based control scheme for calibrated cameras obeying the unified projection model is proposed. The proposed decoupled scheme is based on the surface of object projections onto the unit sphere. Such features are invariant to rotational motions. This allows the control of translational motion independently from the rotational motion. Finally, the proposed results are validated with experiments using a classical perspective camera as well as a fisheye camera mounted on a 6 *dofs* robot platform.

I. INTRODUCTION

In image based visual servoing (IBVS), the control of the camera position is performed by canceling the feature errors in the image [18]. This yields some degree of robustness to disturbances as well as to calibration errors. On the other hand, if the initial error between the initial and the desired positions is large, IBVS may produce erratic behavior such as convergence to local minima and an inappropriate camera trajectory due to coupling between the controlled *dofs* [1]. Usually, IBVS is considered as a servoing approach suitable only for "small" displacements so a basic idea consists of sampling the initial errors in order to ensure that the error at each iteration remains small in order to overcome the IBVS problems already mentioned. That is, using a path planning step jointly with the servoing one[12][4].

The main cause of trouble for IBVS is the strong non-linearities in the relation from the image space to the workspace which are generally observed in the interaction matrix. In principle, an exponential decoupled decrease would be obtained simultaneously on the visual features and on the camera velocity (perfect behavior) if the interaction matrix was constant, which is unfortunately not the case. To overcome the non-linearity problem, the approximation can be improved by incorporating second order terms (based on the Hessian [8], for instance). Another approach consists of selecting features with good decoupling and linearizing properties. In fact, the choice of feature directly influences the closed-loop dynamics in task-space. In [3] features including the distance between two points in the image plane and the orientation of the line connecting those two points was proposed. In [18] the relative area of two projected surfaces has been proposed as a feature. In [13],

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 to obtain some decoupling properties. For the same object, six visual features have been designed in [2] to control the six dofs of a robot arm, following a partitioned approach. In [7], 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 [6], 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. In [10], Mahony *et al* deal with the selection of the optimal feature to control the camera motion with respect to the depth axis. Tatsambon *et al* in [17] proposed a decoupled visual servoing from spheres using a spherical projection model. Despite of the large quantity of results obtained in the last few years, the choice of the set of visual features to be used in the control scheme is still an open question in terms of stability analysis and validity for different kinds of sensor and environment.

The results presented in this paper belong to a series of research about the use of invariants for decoupled image based visual control. More precisely, the invariance property of some combinations of image moments computed from image regions or a set of points are used to decouple the degrees of freedom from each-other. For instance, in [14], [15], 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 [15]. For instance, using such features, the interaction matrix block corresponding to the translational velocity can be a constant block diagonal. However, these works only concerned planar objects and conventional perspective cameras. More recently, a new decoupled image-based control scheme from the projection onto a unit sphere has been proposed in [16]. The proposed method is based on polynomials invariant to rotational motion computed from a set of image points. In this paper, we propose a more generic and efficient decoupled scheme valid when the object is defined by set of points as well as by image regions (or closed contour). The proposed features also reduce the sensitivity of interaction matrix entries to object depth distribution.

In the next section we recall the unified camera model. In Section III, theoretical details about feature selection and obtaining the interaction matrices are detailed. A new vector

O. Tahri is with ISR, University of Coimbra, Polo II PT-3030-290 Coimbra, Portugal omartahri@isr.uc.pt
Y. Mezouar is with LASMEA, Université Blaise Pascal, 63177 AUBIERE, France. youcef.mezouar@lasmea.univ-bpclermont.fr
F. Chaumette is with INRIA Rennes Bretagne Atlantique, France. francois.chaumette@irisa.fr
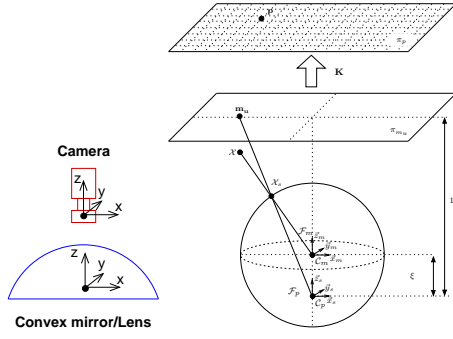P. Corke is with CSIRO ICT Centre, PO Box 883 Kenmore 4069, Australia Peter.Corke@csiro.au

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

of six features to control the six camera degrees of freedom is proposed. Finally, in Section IV, experimental results obtained using conventional camera and a fisheye camera mounted on a *6 dofs* robot are presented to validate our approach.

II. CAMERA MODEL

Central imaging systems can be modeled using two consecutive projections: spherical then perspective. This geometric formulation called the *unified model* was proposed by Geyer and Daniilidis in [5]. Consider a virtual unitary sphere centered on C_m and the perspective camera centered on C_p. The frames attached to the sphere and the perspective camera are related by a simple translation of $-\xi$ along the Z-axis. Let $\mathcal{X}$ be a 3D point with coordinates $\mathcal{X} = [X\ Y\ Z]$ in $\mathcal{F}_m$. The world point $\mathcal{X}$ is projected on to the image plane at a point with homogeneous coordinates $\mathbf{p} = \mathbf{K}\mathbf{m}$, where $\mathbf{K}$ is a 3×3 upper triangular matrix containing the conventional camera intrinsic parameters coupled with mirror intrinsic parameters and

$$\mathbf{m} = \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} \frac{X}{Z+\xi\|\mathcal{X}\|} & \frac{Y}{Z+\xi\|\mathcal{X}\|} & 1 \end{bmatrix} \quad (1)$$

The matrix $\mathbf{K}$ and the parameter ξ can be obtained after calibration using, for example, the methods proposed in [11]. In the sequel, the imaging system is assumed to be calibrated. In this case, the inverse projection onto the unit sphere can be obtained by:

$$\mathcal{X}_\mathbf{s} = \lambda \begin{bmatrix} x & y & 1-\frac{\xi}{\lambda} \end{bmatrix} \quad (2)$$

where $\lambda = \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{1+x^2+y^2}$

Note that the conventional perspective camera is nothing but a particular case of this model (when $\xi = 0$). The projection onto the unit sphere from the image plane is possible for all sensors obeying the unified model.

III. THEORETICAL BACKGROUND

In this Section, the theoretical background of the main idea of this work will first be detailed. Then, six new features will be proposed to control the six *dofs* of the robot-mounted camera.

A. Invariants to rotational motion from a projection onto a sphere

The decoupled control we propose is simply based on the invariance property of the projection shape of an object onto a sphere under rotational motion. In this way, the following invariant polynomial to rotations has been proposed in [16] to control the translational *dofs*:

$$I_1 = m_{200}m_{020} - m_{200}m_{002} + m_{110}^2 + m_{101}^2 - m_{020}m_{002} + m_{011}^2 \quad (3)$$

where:

$$m_{i,j,k} = \sum_{h=1}^{N} x_{s_h}^i\ y_{s_h}^j\ z_{s_h}^k \quad (4)$$

$(x_s,\ y_s,\ z_s)$ being the coordinates of a 3D point. In our application, these coordinates are nothing but the coordinates of a point projected onto the unit sphere. This invariance to rotations is valid whatever the object shape and orientation. In this paper, the surface Δ of the object projection onto a sphere will be used instead of the polynomial mentioned above to control the translational *dofs*. In fact, the surface of the object projection onto a sphere is nothing but the moment of order 0 that can be computed using the general formula:

$$m_{s_{i,j,k}} = \iint_{region} x_s^i y_s^j z_s^k ds \quad (5)$$

The surface is a generic descriptor that can be computed from an image region defined by a closed and complex contour or simply by a polygonal curve. Furthermore, as it will be shown, after an adequate transformation, a new feature can be obtained from the projection surface such that the corresponding interaction matrix is almost constant with the depth distribution. In the remainder of this paper, the surface of the triangles built by the combination of three non-collinear points (from a set of N points) will be considered.

For planar objects and triangle is of course a planar object, it has been shown that the interaction matrix related to the moment can be obtained by [14]:

$$\mathbf{L}_{m_{s_{i,j,k}}} = \begin{pmatrix} m_{s_{vx}} & m_{s_{vy}} & m_{s_{vz}} & m_{s_{wx}} & m_{s_{wy}} & m_{s_{wz}} \end{pmatrix} \quad (6)$$

where :

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

where $\beta = i+j+k+3$ and $(A,\ B,\ C)$ are the parameters defining the object plane in the camera frame:

$$\frac{1}{r} = Ax_s + By_s + C \quad (7)$$

From (6) we can show that $m_{s_{wx}} = m_{s_{wy}} = m_{s_{wz}} = 0$ when $i = j = k = 0$, thus the feature $\Delta = m_{000}$ is invariant to rotational motions.

B. Variations of the interaction matrix related to the surface with respect to the camera position

As has been mentioned above, the problems observed using IBVS are in general due to the strong variations of the interaction matrix with respect to camera position. Therefore, one of the main goals of this work is to decrease these variations. Note firstly that designing a decoupled or a partitioned system is a step toward this goal, since it introduces terms equal to 0 in the interaction matrix. In the following, a transformation is proposed to decrease the variation of the interaction matrix with respect to the object depth.

1) Variation with respect to translational motion: In [10], using a square object, it was shown that for good z-axis behavior in IBVS, one should choose image features that scale as $s \sim z$ (z is the object depth). In [15], the same idea is extended to any object shape using bi-dimensional moments. Using the conventional perspective projection model, the selected feature is $s = \frac{1}{\sqrt{m_{00}}}$ in the case where the object is defined by an image region. m_{00} is the bi-dimensional moment of order 0 (that is the object surface in the image). In the case where the object is defined by a set of discrete points, the selected optimal feature is $s = \frac{1}{\sqrt{(\mu_{20}+\mu_{02})}}$, where μ_{ij} are the central moments computed from a set of discrete points (see [15], for more theoretical details). In fact, the selected features allows us to obtain an interaction matrix that changes slowly with respect to depth (and is even constant if the object is parallel to the image plane). We now show that, the behavior of the surface Δ of an object projection onto the unit sphere is similar ($\Delta \sim \frac{1}{z^2}$ and $\frac{1}{\sqrt{\Delta}} \sim z$).

Let $\mathbf{L}_\Delta = [L_x, L_y, L_z, 0, 0, 0]$ and $\mathbf{L}_{\frac{1}{\sqrt{\Delta}}} = [L_{x_1}, L_{y_1}, L_{z_1}, 0, 0, 0]$ be the interaction matrices related to the projection surface Δ and $\frac{1}{\sqrt{\Delta}}$ respectively. From (6), it can be obtained that:

$$\begin{cases} L_x = A(3m_{200} - m_{000}) + 3Bm_{110} + 3Cm_{101} \\ L_y = 3Am_{110} + B(3m_{020} - m_{000}) + 3Cm_{011} \\ L_z = 3Am_{101} + 3Bm_{011} + C(3m_{002} - m_{000}) \\ L_{x_1} = -\frac{L_x}{2\Delta\sqrt{\Delta}}, \; L_{y_1} = -\frac{L_y}{2\Delta\sqrt{\Delta}}, \; L_{z_1} = -\frac{L_z}{2\Delta\sqrt{\Delta}} \end{cases} \quad (8)$$

It can be shown that the choice of $s = \frac{1}{\sqrt{\Delta}}$ is better than the choice $s = \Delta$. To illustrate this, Figure 3 gives the variation of the interaction matrix entries with respect to translational motion applied to the following triangle in the unit sphere frame:

$$\mathcal{X} = \begin{bmatrix} -0.15 & -0.15 & 0.3 \\ 0.2598 & -0.2598 & -0. \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \quad (9)$$

The triangle shape is given on Figure 2.a.

The results presented in Figure 3 correspond to configurations where $A = B = 0$ and $m_{101} = m_{010} = 0$. Plugging all into (8) we can obtain $L_x = L_{x_1} = L_y = L_{y_1} = 0$. Indeed, from Figures 3.(a) and 3.(b), it can be seen that $L_x = L_{x_1} = L_y = L_{y_1} = 0$ whatever the object depth is. In practice, the features Δ and $\frac{1}{\sqrt{\Delta}}$ depend mainly on the translational motion along z-axis. From Figures 3.(a) and
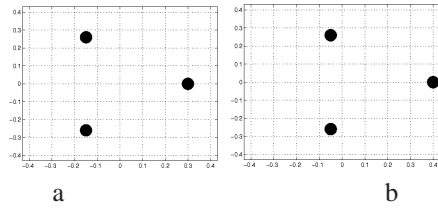

Fig. 2.   Triangle shapes

3.(b), it can also be seen that $L_{z_1} = -C(3m_{002}-\Delta)/(2\Delta^{\frac{3}{2}})$ is almost constant and largely invariant to the object depth. On the other hand $L_z = C(3m_{002}-\Delta)$ decreases to 0 when the object depth increases.

The variation of interaction matrix elements for translational motion with respect to x-axis and y-axis motion are given in Figures 3.(c) to 3.(f). Firstly, it can be seen that x-axis translational motion influences mainly the entries corresponding to the x-axis and z-axis. In the same way, y-axis translational motion influences mainly the entries corresponding to the y-axis and z-axis. Furthermore, variation of the interaction matrix entries for x-axis and y-axis translational motion are more uniform for $\frac{1}{\sqrt{\Delta}}$ than for Δ.
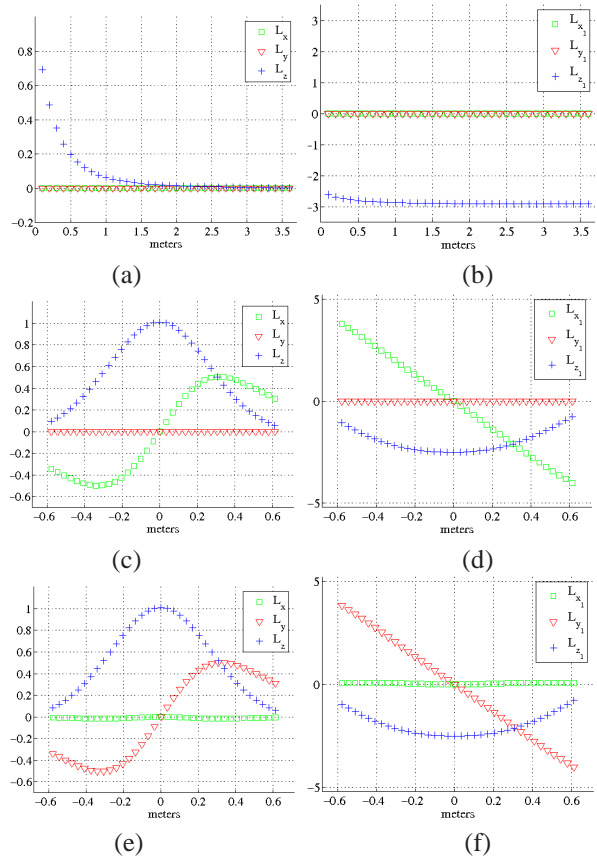

Fig. 3.   Results obtained for $s = \Delta$ ((a) variation with respect to depth, (c) variation with respect to x-axis translation (e) variation with respect to y-axis translation); Results obtained for $s = \frac{1}{\sqrt{\Delta}}$ ((b) variation with respect to depth, (d) variation with respect to x-axis translation (f) variation with respect to y-axis translation)

2) Variations with respect to the camera frame orientation: Despite the fact that the surface of the projection of a target onto a sphere is invariant to rotations, its related interaction matrix depends naturally on the camera frame orientation. In order to explain that, let us consider two frames $\mathcal{F}_1$ and $\mathcal{F}_2$ related to the unit sphere with different orientations ($^1\mathbf{R}_2$ is the rotation matrix between the two frames) but with the same center. In this case, the value of the projection surface onto the sphere is the same for the two frames, since it is invariant to rotational motions. Now, let us consider that a translational velocity $\mathbf{v}_1$ is applied to the frame $\mathcal{F}_1$. This is equivalent to applying a translational velocity to the frame $\mathcal{F}_2$ but taking into account the change of frame ($\mathbf{v}_2 = {}^1\mathbf{R}_2\mathbf{v}_1$). Since the interaction matrix links the features variation with the velocities (i.e. $\dot{\mathbf{s}} = \mathbf{L}_s\mathbf{v}$), the interaction matrix for the frame $\mathcal{F}_2$ is nothing but the interaction matrix computed for the frame $\mathcal{F}_1$ by the rotation matrix $^1\mathbf{R}_2$. This result shows that rotational motions do not change the rank of the interaction matrix of the features used to control the translational *dofs*.

C. Features selection

As in [16], we could consider the center of gravity of the object's projection onto the unit sphere to control the rotational degrees of freedom:

$$\mathbf{x_{s_g}} = (\ x_{s_g}, \quad y_{s_g}, \quad z_{s_g}\ ) = (\ \frac{m_{100}}{m_{000}}, \quad \frac{m_{010}}{m_{000}}, \quad \frac{m_{001}}{m_{000}}\ )$$

In fact, only two coordinates of $\mathbf{x_{s_g}}$ are useful for the control since $\mathbf{x_{s_g}}$ belongs to the unit sphere making one coordinate dependent.

That is why in order to control rotation around the optical axis, the mean orientation of all segments in the image is used as a feature. Each segment is built using two different points in a geometrically correct image. In the case where the objects are defined by contours rather than simple triangles, the object orientation in the image can be used as in [15] for instance.

Finally, as mentioned previously, the invariants to 3D rotation will be considered to control the translation. For the reason mentioned above it is $s = \frac{1}{\sqrt{\Delta}}$ that will be used to control the translational motions instead of $s = \Delta$. In practice, three different targets (i.e. three different triangles or three different contours) such that their centers are non-collinear might be enough to control the three translational *dofs*. In the next section, experimental results are presented to validate these theoretical results.

IV. EXPERIMENTAL RESULTS

In this section, simulations results are firstly presented using four non coplanar points. Thereby, a series of experiments using two kinds of camera (conventional and fisheye) will be shown.

A. Simulation results using 3D objects

In these simulations, the set of points is composed of 4 non coplanar points. The desired position corresponds to the

3D points coordinates defined in the camera frame as follow:

$$\mathbf{X_d} = \begin{pmatrix} 0 & -0.2 & 0 & 0.2 \\ 0.2 & 0 & -0.2 & 0 \\ 0.9 & 1. & 1 & 1.2 \end{pmatrix} \qquad (10)$$

In the first simulation, only the rotational motion given by (11) has been considered. The corresponding results are given on Figure 4. From Figure 4.a, it can be seen that a nice decrease of the features errors is obtained. Furthermore, since the considered translational motion is null, the translational velocity computed using the invariants to rotations are null also (see Fig. 4.b). If the point Cartesian coordinates were used to control the camera position, as in classical IBVS, an undesired and strong translational motion with respect to the optical axis would have been obtained [16], [1]. Finally, Figure 4.c shows good behavior of the rotational velocities despite the large rotational displacement to perform between the desired and the initial camera positions.

$$\boldsymbol{\theta}\mathbf{u} = \begin{bmatrix} -7.90 & 23.70 & 158.0 \end{bmatrix}^o \qquad (11)$$

In the second simulation, a generic motion combining the rotational motion given by (11) and the following translational motion has been considered:

$$\mathbf{t_1} = \begin{bmatrix} -0. & -0.3 & 1 \end{bmatrix} \qquad (12)$$

The obtained results are given on Figs. 4.d, 4.e and 4.f. Despite the large motion, it can be seen that a satisfactory behavior is obtained for the feature errors (see Fig 4.d). Furthermore, similar satisfactory behaviors are simultaneously obtained for the velocities (see Figs 4.e and 4.f). From these plots, it can be also seen that the behavior of the rotational motion is still almost identical to the behavior obtained when only a rotational motion was considered (compare 4.e and 4.b), thanks to the efficient decoupling obtained using the invariant to rotations.

B. Experimental validations results using a conventional and a fisheye cameras

For all these experiments, only approximations of the point depths for the desired position are used. More precisely, the interaction matrices are computed using the current values of the points in the image and constant approximated desired point depths.

1) Results using a conventional perspective camera: In a first experiment, only a rotational motion around the camera optical axis ($80dg$) has been considered between the initial and the desired camera positions. The desired image and the current one are given respectively on Figures 5.a and 6.a. Four combinations of triangles obtained from the four point target are used to control the translational motion. The obtained results are given on Figures 6.b, 6.c and 6.d. From 6.b, it can be seen that a nice decrease of the features errors is obtained. Furthermore, from Fig. 6.b, since the considered translational motion is null, the translational velocity computed using the invariants to rotations are almost

null. The observed small translational velocities are due to the weak calibration of the camera. Finally, Fig. 6.d shows good behavior of the rotational motions.

In a second experiment using a conventional camera, a complex motion is considered between the initial and the desired camera positions. The same desired camera position as for the first experiment is used. The image corresponding to the initial position of the camera is given in Figure 7.a. From Figures 7.b, it can be noticed that the feature errors behavior is very satisfactory, despite the errors in camera calibration and points depth (the point depths are not computed at each iteration). The same satisfactory behavior is obtained for translational and rotational velocities (see Figures 7.c and 7.d). Indeed, nice decreases of the feature errors as well as for the velocities are obtained.

2) Results using a fisheye camera: As for the conventional camera, only a rotational motion around the camera optical axis ($80dg$) has been considered first between the initial and the desired camera positions. The desired image and the current one are given respectively in Figures 5.b and 8.a. The obtained results are given in Figures 8.b, 8.c and 8.d. From these figures, it can be noticed that the translational velocity computed using the invariants to rotations are almost null, as for the conventional camera. The obtained results for the rotational velocities as well as for the feature errors is also similar to those obtained using the conventional camera.

In the last experiment, a generic motion combining rotational motion and translational one is considered between the initial and the desired positions. The image corresponding to the desired position is given in Figure 5.b. The image corresponding to the initial position is given in Figure 9.a. From Figures 9.b, 9.c and 9.d, it can be seen that a satisfactory behavior is obtained using the proposed features. As with the conventional camera, a nice decrease of the features errors as well as of the velocities is also obtained using a fisheye camera.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, a generic decoupled image-based control using the projection onto the unit sphere was proposed. More precisely, the surface of the projections of objects onto the sphere were used to independently control the translational motion from the rotational motion. Firstly, the proposed decoupled control is valid for all cameras obeying the unified camera model. Further, it is also valid for objects defined by closed contours (3 contours at least) as well as by a set of points. The proposed features allows also to decrease significatively the variations of the interaction matrix with respect to the camera positions. Finally, the controller has been experimentally validated and results presented using two kinds of camera: conventional and fisheye. Both planar and non planar target have been used for validations results. Future works will be devoted to extend these results to the pose estimation problem.

REFERENCES

[1] F. Chaumette. Potential problems of stability and convergence in imagebased and position-based visual servoing. In Springer-Verlag,
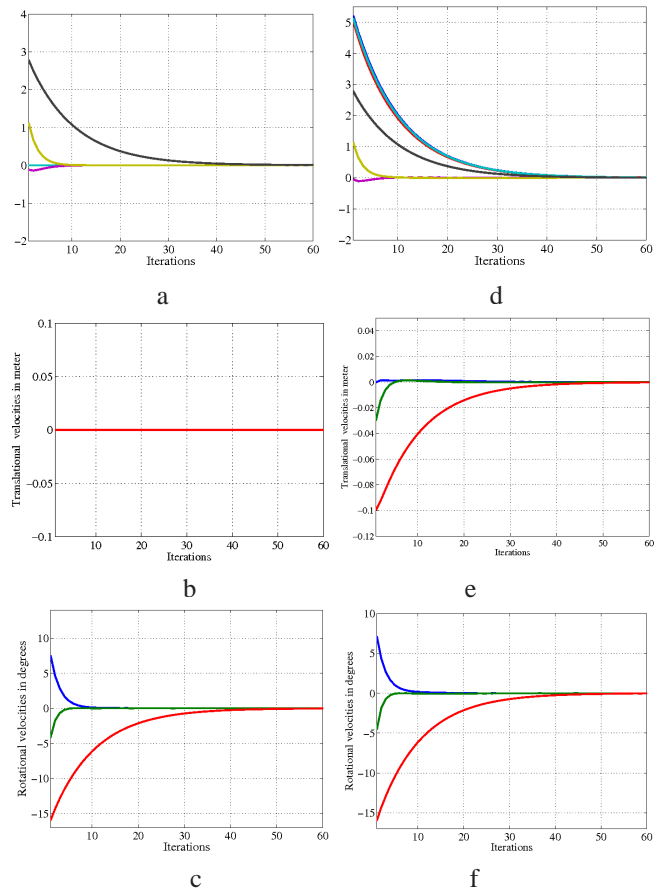
Fig. 4. Left: Results for large pure rotational motion (a) feature errors, b) translational velocities(m/s), c) rotational velocities (degree/s)), right: results for large general motion (d) feature errors, e) translational velocities (m/s), f) rotational velocities (degree/s))
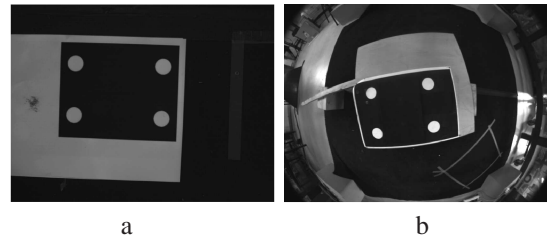

Fig. 5. Desired images: a) conventional camera, b) fisheye camera

editor, *The Confluence of Vision and Control*, volume 237 of *LNCIS*, pages 66–78, 1998.
[2] 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.
[3] J. Feddema and O. Mitchell. Vision-guided servoing with feature-based trajectory generation. 5(5):691–700, October 1989.
[4] D. Fioravanti, B. Allotta1, and A. Rindi1. Image based visual servoing for robot positioning tasks. *Meccanica*, 43(3):291–305, June 2008.
[5] C. Geyer and K. Daniilidis. Mirrors in motion: Epipolar geometry and motion estimation. *International Journal on Computer Vision*, 45(3):766–773, 2003.
[6] 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.
[7] 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*,
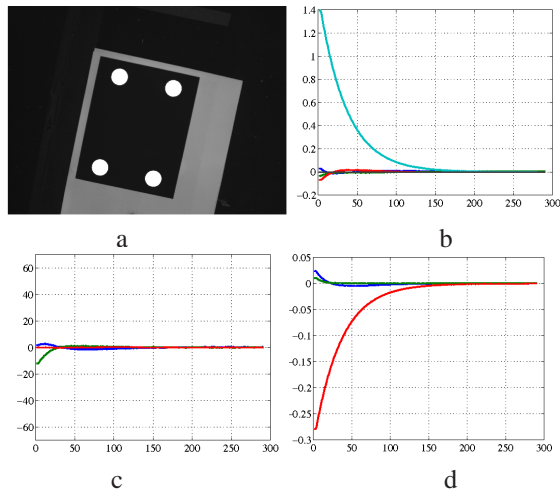
Fig. 6. Results for pure rotation motion ($80dg$) using conventional camera: a) initial image, b) feature errors, c)translational velocities(mm/s), d) rotational velocities (rad/s)
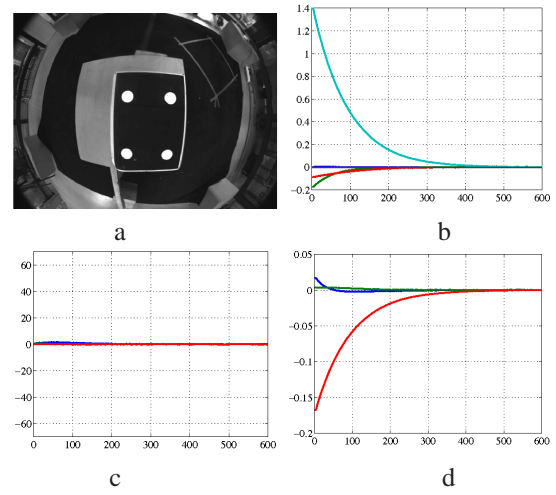

Fig. 8. Results for pure rotation motion ($80dg$) using fisheye camera: a) initial image, b) feature errors, c)translational velocities(mm/s), d) rotational velocities (rad/s)
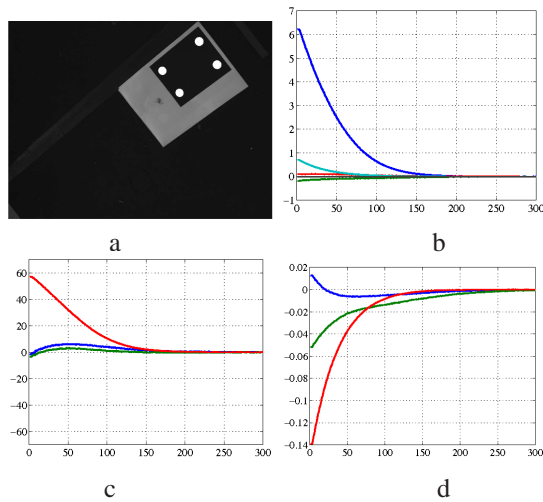

Fig. 7. Results for complex motion using conventional camera: a) initial image, b) feature errors, c)translational velocities(mm/s), d) rotational velocities (rad/s)
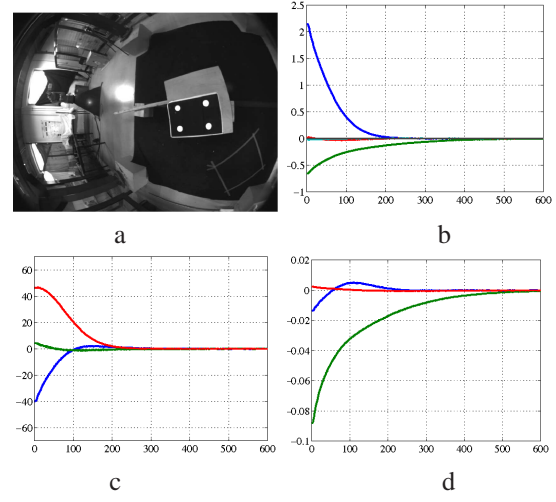

Fig. 9. Results for complex motion using fisheye camera:a) initial image, b) feature errors, c) translational velocities(mm/s), d) rotational velocities (rad/s)

Lausanne, Switzerland, October 2002.

[8] J. T. Lapreste and Y. Mezouar. A hessian approach to visual servoing. In *International Conference on Intelligent Robots and Systems*, pages 998–1003, Sendai, Japan, September 28 October 2 2004.

[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, ICRA'99*, pages 269–274, Detroit, Michigan, May 1999.

[10] R. Mahony, P. Corke, and F. Chaumette. Choice of image features for depth-axis control in image-based visual servo control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'02*, volume 1, pages 390–395, Lausanne, Switzerland, October 2002.

[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] 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, ICRA'04*, pages 255–260, New Orleans, Louisiana, April 2004.

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

[15] 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.

[16] O. Tahri, F. Chaumette, and Y. Mezouar. New decoupled visual servoing scheme based on invariants from projection onto a sphere. In *In IEEE International Conference on Robotics and Automation, ICRA'08*, Pasadena, California, USA, May 19-23 2008.

[17] 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.

[18] L. Weiss, A. C. Sanderson, and C. P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal on Robotics and Automation*, RA-3(5), October 1987.