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

Omar Tahri, Youcef Mezouar, François Chaumette and Peter Corke

***Abstract*—This paper proposes a generic decoupled image-based control scheme for cameras obeying the unified projection model. The scheme is based on the spherical projection model. Invariants to rotational motion are computed from this projection and used to control the translational degrees of freedom. Importantly we form invariants which decrease the sensitivity of the interaction matrix to object depth variation. Finally, the proposed results are validated with experiments using a classical perspective camera as well as a fisheye camera mounted on a 6-DOF robotic platform.**

***Index Terms*—visual servoing, decoupling, invariants, omnidirectional cameras.**

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. Image features that can be, and have been, used include the coordinates of interest points, the parameters of lines or conics, and the moments of planar patches. Others are possible. We wish to choose features that lead to control behavior that is optimal with respect to the image, for instance keeping points within the field of view, and also with respect to 3D camera motion, for instance minimal distance moved and avoidance of robot singularities.

The choice of visual features has a strong influence on the performance of the control system and on the ability to analyze the system's dynamics. A common approach using simple features is image point-based visual servoing. While theoretically suitable only for "small" displacements in practice it is quite robust but less than optimal in terms of 3D motion [3].

One way to improve performance is to sample the initial errors to ensure that the error at each iteration remains small in order to overcome the problems previously mentioned. This combines a path planning process jointly with the servoing one [21], [9], [23], [5], [4]. A second way involves more modeling of the non-linearities in the relationship between the image and workspace. Lapreste *et al* in [15] present a method for estimating the control matrix in visual servoing using a second-order approximation of the projection function based on a Hessian approximation. The main drawback is that this method introduces a number of supplementary parameters. In order to avoid the Hessian computation, an efficient method combining the desired and the current values of the interaction matrix is proposed in [18]. The latter method has been improved in [28] by taking into account the tensor change of frames.

We are interested in approaches that consider performance measures to choose visual features with good decoupling and linearizing properties. In fact, the choice of features directly influences the closed-loop dynamics in task-space. Several works have been realized in image-based visual servoing following the same general objective. In [8], features including the distance between two points in the image plane and the orientation of the line connecting those two points was proposed. In [31] the relative area of two projected surfaces was proposed as a feature. In [22], a vanishing point and the horizon line were selected which ensures good decoupling between translational and rotational degrees of freedom (DOFs). In [16], vanishing points have also been used for a dedicated object (a rectangle), once again to obtain some decoupling properties. For the same object, six visual features have been designed in [6] to control the six DOFs of a robot arm, following a partitioned approach. In [14], 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 [13], 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 [17], 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 [30] propose a decoupled visual servoing from spheres using a spherical projection model. Despite many reported results 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. Performance criteria for choosing feature sets include stability and validity for different kinds of sensors and environments.

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 appropriate combinations of moments. Moment invariants have been well studied for pattern recognition, see [25], [19], [12] for instance. This invariance property is also of particular interest in visual servoing. By selecting an appropriate combination of moments, it becomes possible to create partitioned systems with good decoupling and linearizing properties [25], [26]. For instance, by using such features, the interaction matrix block corresponding to the translational velocity can be a block diagonal with no depth dependence. However this approach is limited to planar objects and conventional perspective cameras. A new decoupled image-based control

Omar Tahri is with Institute for Systems and Robotics, Polo II 3030-290 Coimbra, Portugal, omartahri@isr.uc.pt

Youcef Mezouar is with LASMEA- Blaise Pascal University, Campus des Cezeaux, 63177 Aubiere, France, mezouar@lasmea.univ-bpclermont.fr

François Chaumette is with INRIA Rennes Bretagne Atlantique, campus de Beaulieu 35042 Rennes Cedex, France, francois.chaumette@irisa.fr

Peter Corke is with School of Engineering Systems, Queensland University of Technology Brisbane, Australia, peter.corke@qut.edu.au

scheme using the projection onto a unit sphere has been proposed in [27] and it is based on polynomials invariant to rotational motion computed from a set of image points. More recently, a decoupled image-based control scheme based on the surface of triangle projection onto a sphere has been proposed in [29]. This paper synthesizes our contributions while developing the theoretical and experimental results. In particular, the computation of the interaction matrix related to the projection surface of triangles is detailed and its invariance to rotations formally shown. This paper also provides a new and complete set of real experiments as well as new simulation results. The proposed control schemes are compared between them but also with an image-based control scheme using points coordinates as visual features.

As mentioned above, the features we propose are computed from the projection onto the unit sphere. This means that the proposed method can work not only with classical perspective cameras but can also be applied to wide-angle cameras obeying the unified model [2], [10]. Wide-angle cameras include catadioptric systems that combine mirrors and conventional cameras to create omnidirectional cameras providing 360^o panoramic views of a scene, or dioptric fisheye lenses [1]. It is highly desirable that such imaging systems have a single viewpoint [1], [24]. That is, there exists a single center of projection, so that every pixel in the sensed images measures the irradiance of the light passing through the same viewpoint in one particular direction. The reason why a single viewpoint is so desirable is that it permits the extension of several results obtained for conventional cameras [11]. In this paper, we also take advantage of the properties of such sensor system to develop control laws that are valid for conventional, catadioptric and fisheye cameras.

In the next section we recall the unified camera model and the control law. Moment definitions and the interaction matrices computation are also presented. In Section III, theoretical details about feature selection are discussed and a new vector of features to control the six camera DOF is proposed. Finally, in Section IV, experimental results obtained using a conventional camera and a fisheye camera mounted on a 6-DOF robot are presented to validate our approach.

II. Modeling

A. 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 [10]. Consider a virtual unitary sphere centered on C_m and the perspective camera centered on C_p (refer to Fig. 1). 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 to

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

and then mapped to the homogeneous image-plane coordinate $\mathbf{p} = \mathbf{K}\mathbf{m}$, where $\mathbf{K}$ is a 3×3 matrix of camera and mirror intrinsic parameters. The matrix $\mathbf{K}$ and the parameter ξ can

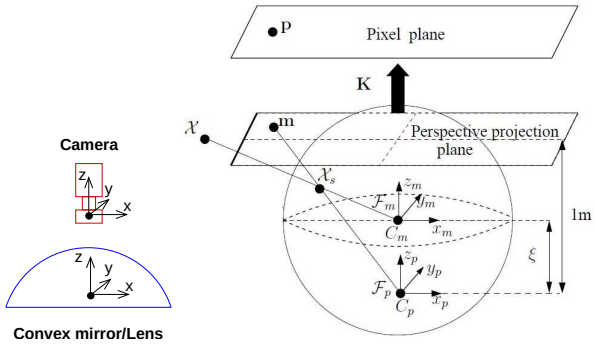


Fig. 1. (left) Catadioptric camera and mirror geometry. (right) Unified image formation.

be obtained after calibration using, for example, the methods proposed in [20]. 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:

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

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 where $\xi = 0$. The projection onto the unit sphere from the image plane is possible for all sensors obeying the unified model.

B. Image-based visual servoing

We define the vector of image features $\mathbf{s}$ and recall that its time variation

$$\dot{\mathbf{s}} = \mathbf{L}_\mathbf{s}\mathbf{V} \quad (3)$$

is linear with respect to the relative camera-object kinematics screw $\mathbf{V} = (\mathbf{v}, \boldsymbol{\omega})$ and $\mathbf{L}_\mathbf{s}$ is the interaction matrix related to $\mathbf{s}$. The control scheme is usually designed to reach an exponential decoupled convergence of the visual features to their desired value $\mathbf{s}^*$ [7]. If we consider an eye-in-hand system observing a static object, the control law is defined as follows:

$$\mathbf{V_c} = -\lambda \widehat{\mathbf{L_s}}^{+} (\mathbf{s} - \mathbf{s}^*) \quad (4)$$

where $\widehat{\mathbf{L_s}}$ is a model or an approximation of $\mathbf{L_s}$, $\widehat{\mathbf{L_s}}^{+}$ the pseudo-inverse of $\widehat{\mathbf{L_s}}$, λ a positive gain, and $\mathbf{V_c}$ the camera velocity sent to the low-level robot controller. Equation (4) is a linear approximation of the non-linear mapping between 3D and image space and therefore valid for small displacements. However for large displacements the approximation is not valid and can lead to suboptimal robot trajectories.

An important issue is therefore to determine those visual features which will allow the system dynamics to be linear over large displacements. Furthermore, using (4) local minima can be reached when the number of features is not minimal. Therefore, one would like to choose a minimal representation (the number of features is equal to the number of DOFs), but without singularities and robust with respect to noise in the image.

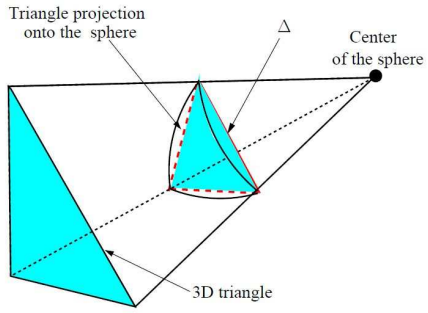


Fig. 2. Triangle projection onto the unit sphere

C. Invariants to rotational motions from the projection onto the surface of unit sphere

The shape of a planar object does not change under rotational motions. After a rotational motion of the sensor frame, it can easily be shown that the projected shape undergoes the same rotational motion as the coordinates of the object 3D points. This means that the invariants to rotation in 3D space are also invariant if the considered points are projected onto the unit sphere. The decoupled features we propose are based on this invariance property. It will be used to select features invariant to rotations in order to control the three translational degrees of freedom. In this way, the following polynomial invariant to rotations has been proposed in [27] to control the translational DOFs:

$$I_1 = m_{200}m_{020} - m_{200}m_{002} + m_{110}^2 + m_{101}^2 - m_{020}m_{002} + m_{011}^2 \tag{5}$$

where $m_{i,j,k}$ is the 3D moment of order $i+j+k$ computed from a discrete set of points defined by the following classical equation:

$$m_{i,j,k} = \sum_{h=1}^{N} x_h^i \; y_h^j \; z_h^k \tag{6}$$

where $(x_h,\ y_h,\ z_h)$ is the coordinates of the h'th point and N is the number of points. In our case, these coordinates are nothing but the coordinates of a point projected onto the unit sphere. In this paper, another kind of invariant is derived from the projection onto the unit sphere. More precisely, the surface that can be computed from the projection of 3 non-collinear points onto the unit sphere will also be used. In this case, two kinds of surfaces invariants to rotations can be defined: the surface defined by the triangle projected onto the unit sphere (defined by three circular arcs corresponding to the projection of the triangle's edges onto the unit sphere) and the surface Δ of the triangle formed by projection onto the sphere, see Figure 2. The latter surface is computed by the well known formula for triangle surface:

$$\Delta = \frac{1}{2} \parallel (\mathcal{X}_{s_2} - \mathcal{X}_{s_1}) \times (\mathcal{X}_{s_3} - \mathcal{X}_{s_1}) \parallel \tag{7}$$

where $\mathcal{X}_{s_1} = (x_{s_1},\, y_{s_1},\, z_{s_1})$, $\mathcal{X}_{s_2} = (x_{s_2},\, y_{s_2},\, z_{s_2})$, $\mathcal{X}_{s_3} = (x_{s_3},\, y_{s_3},\, z_{s_3})$ are the coordinates of the triangle's vertices projected onto the unit sphere.

In the following it is this surface Δ that will be used. We will show that after an adequate transformation new features can be obtained from Δ, and also from I_1 given by (5), such that the corresponding interaction matrices are almost constant with respect to variation of object depth. A comparison of the use of the two features will be made.

D. Interaction matrix

In the case of moments computed from a discrete set of points, the derivative of (6) with respect to time is given by:

$$\begin{aligned}\dot{m}_{i,j,k} = \sum_{h=1}^{N} (&i\; x_{s_h}^{i-1}\; y_{s_h}^{j}\; z_{s_h}^{k} \dot{x}_{s_h} + j\; x_{s_h}^{i}\; y_{s_h}^{j-1}\; z_{s_h}^{k} \dot{y}_{s_h} \\ &+ k\; x_{s_h}^{i}\; y_{s_h}^{j}\; z_{s_h}^{k-1} \dot{z}_{s_h})\end{aligned} \tag{8}$$

The interaction matrix $\mathbf{L}_{\mathcal{X}_s}$ for a point on the unit sphere is well known, refer to [13], [25], [30].

$$\mathbf{L}_{\mathcal{X}_s} = \begin{bmatrix} -\frac{1}{r}\mathbf{I_3} + \frac{1}{r}\mathcal{X}_s\mathcal{X}_s^\top & [\mathcal{X}_s]_\times \end{bmatrix} \tag{9}$$

where r is the distance of the 3D point to the sphere center. For any set of points (coplanar or non-coplanar), we can combine that interaction matrix with that related to $\mathbf{L}_{\mathbf{m_{i,j,k}}}$, from (8), to obtain:

$$\mathbf{L}_{m_{i,j,k}} = \begin{bmatrix} m_{vx} & m_{vy} & m_{vz} & m_{wx} & m_{wy} & m_{wz} \end{bmatrix} \tag{10}$$

where:

$$\begin{cases} m_{vx} = \displaystyle\sum_{h=1}^{N} \frac{-i x_{s_h}^{i-1}\; y_{s_h}^{j}\; z_{s_h}^{k} + \beta_d x_{s_h}^{i+1} y_{s_h}^{j} z_{s_h}^{k}}{r_h} \\ m_{vy} = \displaystyle\sum_{h=1}^{N} \frac{-j x_{s_h}^{i}\; y_{s_h}^{j-1}\; z_{s_h}^{k} + \beta_d x_{s_h}^{i} y_{s_h}^{j+1} z_{s_h}^{k}}{r_h} \\ m_{vz} = \displaystyle\sum_{h=1}^{N} \frac{-k x_{s_h}^{i}\; y_{s_h}^{j}\; z_{s_h}^{k-1} + \beta_d x_{s_h}^{i} y_{s_h}^{j} z_{s_h}^{k+1}}{r_h} \\ m_{wx} = j m_{i,j-1,k+1} - k m_{i,j+1,k-1} \\ m_{wy} = k m_{i+1,j,k-1} - i m_{i-1,j,k+1} \\ m_{wz} = i m_{i-1,j+1,k} - j m_{i+1,j-1,k} \end{cases}$$

and $\beta_d = i + j + k$. In the particular case of a coplanar set of points, the interaction matrix related to $m_{i,j,k}$ can be determined [25]:

$$\begin{cases} m_{vx} = A(\beta_d m_{i+2,j,k} - i m_{i,j,k}) \\ \qquad + B(\beta_d m_{i+1,j+1,k} - i m_{i-1,j+1,k}) \\ \qquad + C(\beta_d m_{i+1,j,k+1} - i m_{i-1,j,k+1}) \\ m_{vy} = A(\beta_d m_{i+1,j+1,k} - j m_{i+1,j-1,k}) \\ \qquad + B(\beta_d m_{i,j+2,k} - j m_{i,j,k}) \\ \qquad + C(\beta_d m_{i,j+1,k+1} - j m_{i,j-1,k+1}) \\ m_{vz} = A(\beta_d m_{i+1,j,k+1} - k m_{i+1,j,k-1}) \\ \qquad + B(\beta_d m_{i,j+1,k+1} - k m_{i,j+1,k-1}) \\ \qquad + C(\beta_d m_{i,j,k+2} - k m_{i,j,k}) \\ m_{wx} = j m_{i,j-1,k+1} - k m_{i,j+1,k-1} \\ m_{wy} = k m_{i+1,j,k-1} - i m_{i-1,j,k+1} \\ m_{wz} = i m_{i-1,j+1,k} - j m_{i+1,j-1,k} \end{cases} \tag{11}$$

where $\alpha = (A,\, B,\, C)$ are the parameters defining the object plane in the camera frame:

$$\frac{1}{r} = \boldsymbol{\alpha}^\top \mathcal{X}_s = Ax_s + By_s + Cz_s \tag{12}$$

The interaction matrix related to Δ can be obtained in a similar way. Let $\mathbf{L_\Delta}$ be the 1×6 interaction matrix related to Δ:

$$\mathbf{L_\Delta} = [\mathbf{L_{\Delta_v}}\ \ \mathbf{L_{\Delta\omega}}] \tag{13}$$

where $\mathbf{L}_{\Delta_v}$ and $\mathbf{L_{\Delta\omega}}$ are respectively two 1×3 matrices that link the time variation of Δ to the translational and the rotational velocities.

Lemma 1. *Δ is invariant to rotations and:*

$$\mathbf{L_{\Delta\omega}} = [\mathbf{0},\ \mathbf{0},\ \mathbf{0}]$$

Proof: Let $\mathbf{L}_{\mathcal{X}_\mathbf{i}}$ be the interaction matrix related to the point $\mathcal{X}_{s_i}$, $\mathbf{L}_{\mathcal{X}_\mathbf{ij}} = \mathbf{L}_{\mathcal{X}_\mathbf{i}} - \mathbf{L}_{\mathcal{X}_\mathbf{j}}$ be the interaction matrix difference and $\mathcal{X}_{ij} = \mathcal{X}_{s_i} - \mathcal{X}_{s_j}$ be the coordinate vector difference. The surface Δ can be written:

$$\Delta = \frac{\sqrt{\mathcal{X}_{31}^\top [\mathcal{X}_{21}]_\times^\top [\mathcal{X}_{21}]_\times \mathcal{X}_{31}}}{2} = \frac{\sqrt{\mathcal{X}_{21}^\top [\mathcal{X}_{31}]_\times^\top [\mathcal{X}_{31}]_\times \mathcal{X}_{21}}}{2} \tag{14}$$

Taking the time derivative of (14), we obtain:

$$\mathbf{L}_\Delta = \frac{\mathcal{X}_{31}^\top [\mathcal{X}_{21}]_\times^\top [\mathcal{X}_{21}]_\times \mathbf{L}_{\mathcal{X}_{\mathbf{31}}} - \mathcal{X}_{31}^\top [\mathcal{X}_{21}]_\times^\top [\mathcal{X}_{31}]_\times \mathbf{L}_{\mathcal{X}_{\mathbf{21}}}}{4\Delta} \tag{15}$$

Combining (15) with (9), it follows that:

$$\mathbf{L}_{\Delta_\omega} = \frac{\mathcal{X}_{21}^\top [\mathcal{X}_{31}]_\times^\top \left([\mathcal{X}_{21}]_\times [\mathcal{X}_{\mathbf{31}}]_\times - [\mathcal{X}_{31}]_\times [\mathcal{X}_{\mathbf{21}}]_\times\right)}{4\Delta} \tag{16}$$

Additionally, it can easily be shown that:

$$[\mathcal{X}_{21}]_\times [\mathcal{X}_{\mathbf{31}}]_\times - [\mathcal{X}_{31}]_\times [\mathcal{X}_{\mathbf{21}}]_\times = [\mathcal{X}_{21} \times \mathcal{X}_{31}]_\times \tag{17}$$

Let us consider $\mathbf{h} = \mathcal{X}_{21} \times \mathcal{X}_{31}$, which allows (16) to be written as:

$$\mathbf{L}_{\Delta_\omega} = \frac{\mathbf{h}^\top [\mathbf{h}]_\times}{4\Delta} \tag{18}$$

from which, we immediately deduce:

$$\mathbf{L}_{\Delta_\omega} = [0\ 0\ 0] \tag{19}$$

which confirms the invariance of Δ to rotations. ■

For translational velocity, after tedious computation, the interaction matrix related to Δ can be written as:

$$\begin{aligned}\mathbf{L}_{\Delta_v} = &\frac{\mathcal{X}_{21}^\top [\mathcal{X}_{31}]_\times^\top [\mathcal{X}_{31}]_\times \boldsymbol{\alpha}^\top (-\mathcal{X}_{21}\mathbf{I_3} + \mathcal{X}_{s_2}\mathcal{X}_{s_2}\mathcal{X}_{s_2}^\top - \mathcal{X}_{s_1}\mathcal{X}_{s_1}\mathcal{X}_{s_1}^\top)}{4\Delta} \\ &+ \frac{\mathcal{X}_{31}^\top [\mathcal{X}_{21}]_\times^\top [\mathcal{X}_{21}]_\times \boldsymbol{\alpha}^\top (-\mathcal{X}_{31}\mathbf{I_3} + \mathcal{X}_{s_3}\mathcal{X}_{s_3}\mathcal{X}_{s_3}^\top - \mathcal{X}_{s_1}\mathcal{X}_{s_1}\mathcal{X}_{s_1}^\top)}{4\Delta}\end{aligned} \tag{20}$$

and after further computations it can be shown that:

$$\mathbf{L}_{\Delta_v} = \mathbf{L}_{\Delta_{v1}} + \mathbf{L}_{\Delta_{v2}} \tag{21}$$

where:

$$\begin{cases} \mathbf{L}_{\Delta_{v_1}} = \dfrac{\mathcal{X}_{31}^\top [\mathcal{X}_{21}]_\times^\top \left[\boldsymbol{\alpha}^\top \mathcal{X}_{21} [\mathcal{X}_{31}]_\times + \boldsymbol{\alpha}^\top \mathcal{X}_{31} [\mathcal{X}_{12}]_\times\right]}{4\Delta} \\ \mathbf{L}_{\Delta_{v2}} = \mathcal{X}_{31}^\top [\mathcal{X}_{21}]_\times^\top \Big[\boldsymbol{\alpha}^\top \mathcal{X}_{s_1} [\mathcal{X}_{32}]_\times \mathcal{X}_{s_1} \mathcal{X}_{s_1}^\top \\ \quad + \boldsymbol{\alpha}^\top \mathcal{X}_{s_2} [\mathcal{X}_{13}]_\times \mathcal{X}_{s_2} \mathcal{X}_{s_2}^\top + \boldsymbol{\alpha}^\top \mathcal{X}_{s_3} [\mathcal{X}_{21}]_\times \mathcal{X}_{s_3} \mathcal{X}_{s_3}^\top\Big] / (4\Delta) \end{cases}$$

In practice, $\mathbf{L}_{\Delta_v}$ depends strongly on $\mathbf{L}_{\Delta_{v_1}}$ because the numerator of $\mathbf{L}_{\Delta_{v_2}}$ is a polynomial of point projections with a higher order than the numerator of $\mathbf{L}_{\Delta_{v_1}}$.

III. FEATURES CHOICE

In this section we detail our choice of image features. Firstly we will explain how to obtain features to control the translational DOFs with interaction matrices that are almost constant with respect to variation in object depth. Then, a vector of features to control all 6 DOF will be proposed.

A. Variation of the interaction matrix with respect to camera pose

As mentioned above, one of the goals of this work is to decrease the system non-linearity and coupling by selecting adequate features. The invariance property, for example, results in some interaction matrix entries being 0, removing coupling between DOF and also being constant during the servoing task. However other entries depend on the camera pose as will be shown next. It will be also shown that the feature choice $I_t = \frac{1}{\sqrt{I_1}}$ and $s_\Delta = \frac{1}{\sqrt{\Delta}}$ leads to interaction matrices that are almost constant with respect to the object depth variation.

1) Variation with respect to rotational motion: Let us consider two frames $\mathcal{F}_\mathbf{1}$ and $\mathcal{F}_\mathbf{2}$ 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 I_t is the same for the two frames, since it is invariant to rotation. Let $\mathcal{X}_s$ and $\mathcal{X}'_s = {}^2\mathbf{R}_1 \mathcal{X}_s$ be the coordinates in the frame $\mathcal{F}_\mathbf{1}$ and $\mathcal{F}_\mathbf{2}$ respectively of a projected point. Let us consider a function invariant to rotations $f(\mathcal{X}_1, \ldots, \mathcal{X}_N)$ that can be computed from the coordinates of N points onto the unit sphere. The invariance condition between the frames $\mathcal{F}_\mathbf{1}$ and $\mathcal{F}_\mathbf{2}$ can be written as:

$$f(\mathcal{X}'_1, \ldots, \mathcal{X}'_N) = f({}^2\mathbf{R}_1 \mathcal{X}_1, \ldots, {}^2\mathbf{R}_1 \mathcal{X}_N) = f(\mathcal{X}_1, \ldots, \mathcal{X}_N) \tag{22}$$

The interaction matrix that links the variation of the function f with respect to translational velocities can be obtained as:

$$\mathbf{L}_{\mathbf{f_v}} = \frac{\partial f(\mathcal{X}_1 + \mathbf{T}, \ldots, \mathcal{X}_N + \mathbf{T})}{\partial \mathbf{T}} \tag{23}$$

where $\mathbf{T}$ is a small translational motion vector. Let us now apply this formula for the camera pose defined by the frame $\mathcal{F}_\mathbf{2}$:

$$\begin{aligned}\mathbf{L}'_{\mathbf{f_v}} &= \frac{\partial f(\mathcal{X}'_1 + \mathbf{T}, \ldots, \mathcal{X}'_N + \mathbf{T})}{\partial \mathbf{T}} \\ &= \frac{\partial f({}^2\mathbf{R}_1 \mathcal{X}_1 + \mathbf{T}, \ldots, {}^2\mathbf{R}_1 \mathcal{X}_N + \mathbf{T}))}{\partial \mathbf{T}}\end{aligned} \tag{24}$$

from which we obtain:

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f\left({}^2\mathbf{R}_1(\mathcal{X}_1 + {}^1\mathbf{R}_2\mathbf{T}), \ldots, {}^2\mathbf{R}_1(\mathcal{X}_N + {}^1\mathbf{R}_2\mathbf{T})\right)}{\partial \mathbf{T}} \tag{25}$$

Combining with the rotational invariance condition (22) we obtain:

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f(\mathcal{X}_1 + {}^1\mathbf{R}_2\mathbf{T}, \ldots, \mathcal{X}_N + {}^1\mathbf{R}_2\mathbf{T})}{\partial \mathbf{T}} \tag{26}$$

leading to

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f(\mathcal{X}_1 + \mathbf{T}', \ldots, \mathcal{X}_N + \mathbf{T}')}{\partial \mathbf{T}'} \frac{\partial \mathbf{T}'}{\partial \mathbf{T}} \tag{27}$$

where $\mathbf{T}' = {}^1\mathbf{R}_2\mathbf{T}$. Finally, combining with (23) yields:

$$\mathbf{L}'_{\mathbf{f_v}} = \mathbf{L}_{\mathbf{f_v}}\ {}^1\mathbf{R}_2 \tag{28}$$

This result was expected since applying a translational velocity $\mathbf{v_1}$ to the frame $\mathcal{F}_1$ is equivalent to applying a translational velocity to the frame $\mathcal{F}_2$ but taking into account the change of frame ($\mathbf{v_2} = {}^\mathbf{2}\mathbf{R_1}\mathbf{v_1}$). This variation is thus natural — the translational velocity applied to the camera frame depends on its orientation. Finally, this result shows that rotational motions do not change the rank of the interaction matrix of the features used to control the translational DOFs. In other words, the rotational motions do not introduce singularities to the interaction matrix and any rank change of the latter depends only on the translational motion.

2) Variation of the interaction matrix with respect to depth: Constant interaction matrix entries are a desirable property and mean that the corresponding features depend linearly of the corresponding DOF. In [17], [26], it was shown that for good z-axis closed-loop behavior in IBVS, one should choose image features that scale as $s \sim Z$ (Z is the object depth) so that the variation of their corresponding interaction matrices with respect to depth is zero. In the case where the object is defined by an image region, the following feature has been proposed to control the motion along and around the optical axis [6], [26]:

$$s_r = \frac{1}{\sqrt{m_{00}}}$$

where m_{00} is the moment of order 0 (object surface in the image) using the conventional perspective projection model. In the case where the object is defined by a set of discrete points, the selected optimal feature for rotation was:

$$s_d = \frac{1}{\sqrt{(\mu_{20} + \mu_{02})}} \tag{29}$$

where μ_{ij} are the central moments computed from a set of discrete points (see [26] for more details). Unfortunately, s_r and s_d only provides invariance to rotations around the optical axis and not to all 3D rotations.

For this reason, $I_t = \frac{1}{\sqrt{I_1}}$ and $s_\Delta = \frac{1}{\sqrt{\Delta}}$ will be used instead of s_d and s_r respectively. To explain this choice, let us first determine how the polynomial invariant I_1 behaves for increasing Z by considering each of its terms. Let us consider the definition of the projection onto the unit sphere:

$$\begin{cases} x_s = \frac{X}{\sqrt{X^2+Y^2+Z^2}} \\ y_s = \frac{Y}{\sqrt{X^2+Y^2+Z^2}} \\ z_s = \frac{Z}{\sqrt{X^2+Y^2+Z^2}} \end{cases} \tag{30}$$

From (30), we can see that if the depth Z increases (assuming $X \ll Z$ and $Y \ll Z$), the point projection coordinates have the following behaviors with respect to depth: $x_s \sim \frac{1}{Z}$, $y_s \sim \frac{1}{Z}$ and $z_s \sim 1$. It follows that: $m_{200} = \sum_{h=1}^{N} x_{s_h}^2 \sim \frac{1}{Z^2}$, $m_{020} = \sum_{h=1}^{N} y_{s_h}^2 \sim \frac{1}{Z^2}$, $m_{110} = \sum_{h=1}^{N} x_{s_h} y_{s_h} \sim \frac{1}{Z^2}$, $m_{101} = \sum_{h=1}^{N} x_{s_h} z_{s_h} \sim \frac{1}{Z}$, $m_{011} = \sum_{h=1}^{N} y_{s_h} z_{s_h} \sim \frac{1}{Z}$ and $m_{002} = \sum_{h=1}^{N} z_{s_h}^2 \sim N$. Neglecting terms depending on $\frac{1}{Z^4}$ the polynomial can be approximated as:

$$I_1 \approx N(m_{200} + m_{020}) - m_{100}^2 - m_{010}^2 \tag{31}$$

Now we can see that $I_1 \sim \frac{1}{Z^2}$ and $I_t = \frac{1}{\sqrt{I_1}} \sim Z$. Note that if the set of points is centered with respect to the optical axis (i.e. $m_{100} = m_{010} = 0$), we have:

$$I_1 \approx N(m_{200} + m_{020}) \tag{32}$$

In this case, note the similarity between $I_t = \frac{1}{\sqrt{I_1}}$ and the features given by (29). In geometric terms, if the set of points is centered with respect to the optical axis, the perspective projections onto the unit sphere and onto the image plane behave in the same way as depth increases. Similarly, it is possible to show that $\Delta \sim \frac{1}{z^2}$ and $\frac{1}{\sqrt{\Delta}} \sim z$. Examples of interaction matrix variation with respect to object depth are given in Section IV-A1. Note finally that the interaction matrix related to s_Δ can be obtained from $\mathbf{L}_\Delta$ as follows:

$$\mathbf{L}_{s_\Delta} = -\frac{1}{2\Delta^{\frac{3}{2}}}\mathbf{L}_\Delta \tag{33}$$

since $\frac{\partial s_\Delta}{\partial t} = -\frac{1}{2\Delta^{3/2}}\frac{\partial \Delta}{\partial t}$. The same applies for I_t. From (33), it is clear that the invariance to rotation shown for Δ and I_1 is still valid for s_Δ and I_t.

B. Feature selection

To control the rotational degrees of freedom consider the center of gravity of the object's projection onto the unit sphere:

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

As in [27], only two coordinates of $\mathbf{x_{s_g}}$ are useful for control since the point projections belongs to the unit sphere making one coordinate dependent. Recall that the interaction related to $\mathbf{x_{s_g}}$ is obtained using (10). In order to control rotation around the optical axis, the mean orientation of all line segments in the image is used as a feature. Each segment is built using two different points in an image obtained by re-projection to a conventional perspective plane. More precisely the segment orientation is defined by:

$$\theta = \operatorname{atan2}(\frac{y_2 - y_1}{d}, \frac{x_2 - x_1}{d}) \tag{34}$$

where x_i and y_i are the coordinates of the points forming the segment in a classical perspective plane and $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ is the distance separating them. By deriving (34) and combining the result with the interaction matrix related to the Cartesian points coordinates, the interaction matrix related to θ is given by [7]:

$$L_\theta = \begin{bmatrix} \theta_{\mathbf{vx}} & \theta_{\mathbf{vy}} & \theta_{\mathbf{vz}} & \theta_{\boldsymbol{\omega x}} & \theta_{\boldsymbol{\omega y}} & -1 \end{bmatrix} \tag{35}$$

where:

$$\begin{cases} \theta_{\mathbf{vx}} = \frac{\sin\theta}{d}(\frac{1}{Z_2} - \frac{1}{Z_1}) \\ \theta_{\mathbf{vy}} = \frac{-\cos\theta}{d}(\frac{1}{Z_2} - \frac{1}{Z_1}) \\ \theta_{\mathbf{vz}} = \frac{(x_2y_1 - x_1y_2)}{d^2}(\frac{1}{Z_2} - \frac{1}{Z_1}) \\ \theta_{\boldsymbol{\omega x}} = \frac{x_1y_1y_2 + x_2y_1y_2 - x_2y_1^2 - x_1y_2^2}{d^2} \\ \theta_{\boldsymbol{\omega y}} = \frac{x_1y_1x_2 + x_1x_2y_2 - y_2x_1^2 - y_1x_2^2}{d^2} \end{cases}$$

and Z_i is the depth of the 3D points.

Finally the invariants to 3D rotation $I_t = \frac{1}{\sqrt{I_1}}$ or $s_\Delta = \frac{1}{\sqrt{\Delta}}$ are considered as features to control the translational DOF. The choice of $I_t = \frac{1}{\sqrt{I_1}}$ or $s_\Delta = \frac{1}{\sqrt{\Delta}}$ is discussed in section IV-E. In practice, when I_t is used, three separate targets such that their centers are non-collinear can be sufficient to control the translational DOFs. In order to ensure the non-singularity of the interaction matrix, the set of points is divided into four subsets (each subset must contain at least 3 points). This allows us to obtain four different features to control the three translational degrees of freedom. Recall that if the set is composed of coplanar points, the simple form (11) of the interaction matrix can be used. If the points are not coplanar, the form (10) has to be used. Similarly, when s_Δ is used, four different triangles can be obtained by combining three non-collinear points among a set of at least four points.

IV. Results

In this section the theoretical results presented above are validated. We will first see how the interaction matrix varies with the depth through two examples. Then, a statistical study of convergence rate using several features is given. Finally, we present the results of a series of experiments with planar and non-planar targets and two kinds of camera (conventional and fisheye).

A. Variation of the interaction matrix with camera pose

1) Variation with respect to camera translation: Fig. 3 shows the variation of the interaction matrix entries related to Δ and $\frac{1}{\sqrt{\Delta}}$ with respect to translational motion applied to the triangle:

$$\mathcal{X} = \begin{bmatrix} -0.15 & -0.15 & 0.3 \\ 0.2598 & -0.2598 & 0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \tag{36}$$

defined in the camera frame. From Figs 3.(a) and 3.(d), it can be seen that $L_x = L_{x_1} = L_y = L_{y_1} = 0$ irrespective of object depth (where $\mathbf{L}_\Delta = [L_x,\ L_y,\ L_z,\ 0,\ 0,\ 0]$ and $\mathbf{L}_{s_\Delta} = [L_{x_1},\ L_{y_1},\ L_{z_1},\ 0,\ 0,\ 0]$). In practice, the features Δ and s_Δ depend mainly on the translational motion with respect to the object axis of view (in this example the axis of view is the camera optical axis). From Fig. 3.(d), it can also be seen that L_{z_1} is almost constant and invariant to the object depth. On the other hand, L_z decreases to 0 when the object depth increases (see Fig. 3.(a)).

The variation of interaction matrix entries for translational motion with respect to x-axis and y-axis motion are given in Figs 3.(b), 3.(c), 3.(e) and 3.(f). Firstly, it can be seen that x-axis translational motion influences mainly the entries corresponding to the x-axis and z-axis. Similarly, 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 s_Δ than for Δ.

As a second example, Fig. 4 gives the variations of the interaction matrix entries of I_1 and $\frac{1}{\sqrt{I_1}}$ with respect to translational motion along the z-axis applied to the four coplanar points defined in the camera frame:

$$\mathbf{X_o} = \begin{pmatrix} -0.3258 & -0.0811 & 0.1487 & 0.2583 \\ -0.0458 & 0.1470 & -0.1052 & 0.0039 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{37}$$

The set of points has been chosen to be approximatively centered with respect to the z-axis ($m_{100} \approx 0$ and $m_{010} \approx 0$). For this reason, it can be seen that $L_x \approx L_{x_1} \approx L_y \approx L_{y_1} \approx 0$ ($\mathbf{L}_{I_1} = [L_x,\ L_y,\ L_z,\ 0,\ 0,\ 0]$ and $\mathbf{L}_{I_t} = [L_{x_1},\ L_{y_1},\ L_{z_1},\ 0,\ 0,\ 0]$). In practice, the features I_1 and $\frac{1}{\sqrt{I_1}}$ also depend mainly on the translational motion with respect to the object axis of view. From Figs 4.(a) and 4.(b), it can also be seen that L_{z_1} is almost constant and largely invariant to the object depth, while L_z decreases to 0 when the object depth increases.

2) Variation with respect to the camera orientation: To illustrate the results presented in Section III-A1, we have computed the variations of the interaction matrix entries related to Δ and $\frac{1}{\sqrt{\Delta}}$ with respect to rotation around the optical axis applied to the triangle defined by the following 3D coordinates in the camera frame:

$$\mathcal{X} = \begin{bmatrix} -0.05 & -0.05 & 0.4 \\ 0.2598 & -0.2598 & 0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \tag{38}$$

These results are shown in Fig. 5. The curves corresponding to the entries L_z and L_{z_1} show that they are constant. This was expected, since a rotational motion around the optical axis does not change the orientation of this axis. Thus, the variation of the selected features with respect to rotational motion around the optical axis remains constant. From the same figure, it can also be seen that the variation of the other entries are sinusoidal functions. In fact, as shown in Section III-A1, the interaction matrix after a rotational motion is the product of this matrix by the rotation matrix.

B. Simulation results using a non planar set of points

In these simulations the target comprises 4 non-coplanar points. The desired position corresponds to the following 3D point coordinates defined in the camera frame:

$$\mathbf{X_d} = \begin{pmatrix} 0 & -0.2 & 0 & 0.2 \\ 0.2 & 0 & -0.2 & 0 \\ 0.9 & 1. & 1 & 1.2 \end{pmatrix} \tag{39}$$

Two different features set are tested to control the translational motion: $s_\Delta = \frac{1}{\sqrt{\Delta}}$ and $I_t = \frac{1}{\sqrt{I_1}}$. From the four points, four different triangles can be obtained. For each triangle, the invariants s_Δ and I_t are used.

In this simulation, we show the advantage of using s_Δ or I_t instead of directly using Δ and I_1. For this purpose, the translational motion $\mathbf{t_0} = (0.2,\ 0.3,\ 0.6)$ meter has been considered between the desired and the initial camera poses. The scalar velocity gain in the control law has been set to $\lambda = 1$. If the system were completely linear, convergence would be obtained in only one iteration. The non-linearity of the system has the effect of damping or magnifying the camera velocities. The results obtained using I_t and I_1 are given in Fig. 6. From Figs 6.(a) and 6.(b), we observe oscillations for

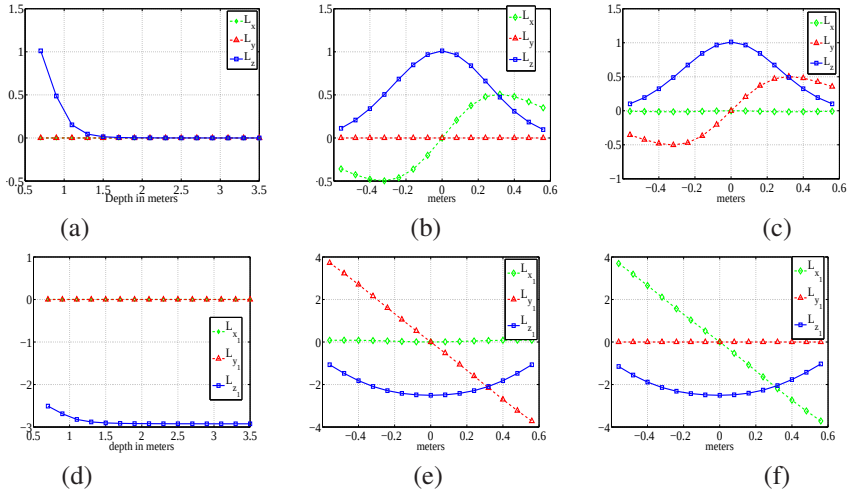


Fig. 3. Results obtained for $s = \Delta$: (a) variation with respect to depth, (b) variation with respect to x-axis translation (c) variation with respect to y-axis translation; Results obtained for $s_\Delta = \frac{1}{\sqrt{\Delta}}$: (d) variation with respect to depth, (e) variation with respect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

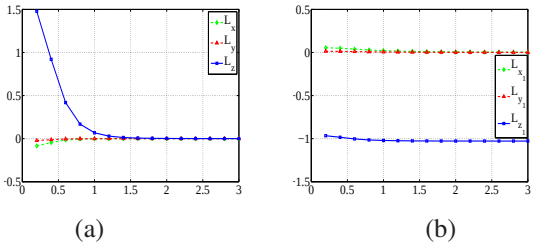


Fig. 4. Variation of the interaction matrix with object depth (in meters): (a) results for I_1, (b) results for I_t

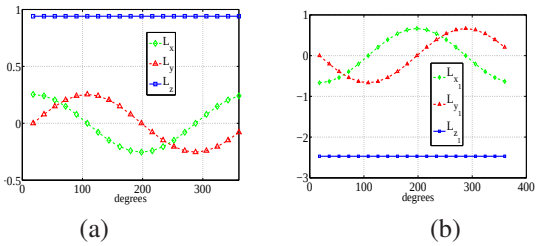


Fig. 5. Variation of the interaction matrix entries with respect to a rotation around the optical axis: (a) result for $s = \Delta$, (b) result for s_Δ

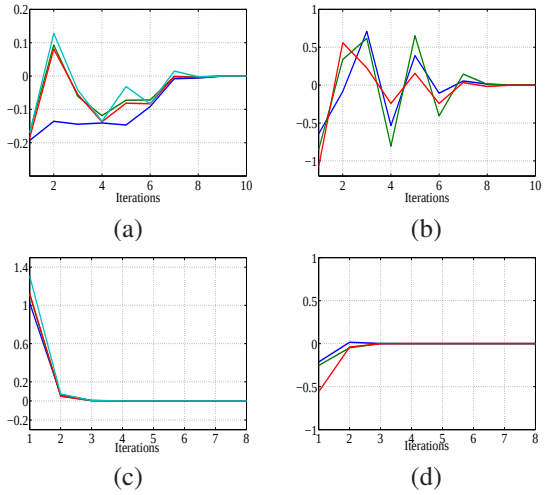


Fig. 6. Results obtained using I_1: (a) features errors, (b) velocities (m/s); obtained results using I_t: (c) features errors, (d) velocities (m/s)

both the feature error and the velocities obtained using I_1 before converging (after 9 iterations). On the other hand, a fast convergence is obtained using I_t without oscillation (after only two iterations the system has almost converged). This shows that using I_t the system has behaved as an almost linear system. The results obtained using s_Δ and Δ are given in Fig. 7. They confirm those obtained using I_t and I_1 — the system converges faster using s_Δ rather than Δ.

In the second simulation, a generic motion combining the rotational motion $\boldsymbol{\theta u} = (-7.90,\ 23.70,\ 158.0)$ degrees and the translational motion $\mathbf{t_1} = (0,\ 0.3,\ 1)$ meters is considered.

The results obtained using s_Δ to control translation are given in Figs. 8.(a), 8.(b) and 8.(c). Despite the large motion, it can be seen that a satisfactory behavior is obtained for the feature errors (see Fig 8.(a)). Similar satisfactory behaviors are simultaneously obtained for the velocities (see Figs 8.(b) and 8.(c)). Furthermore, from the obtained results (refer to the plot corresponding to the translational velocities), it can be seen that one of the velocities (translation with respect to x-axis) is null, which means that the considered displacements with respect to y-axis and z-axis do not produce any velocity with respect to x-axis. This confirms the decoupling properties of our control. Finally, the obtained results using I_t to control the translational motions are given in Figs. 8.(d), 8.(e) and 8.(f). From these figures, it can be seen that the behavior is satisfactory and almost identical to the one obtained using s_Δ.

C. Convergence rate and robustness

In this part, comparison of the visual servoing using the features I_t, s_Δ and the Cartesian image point coordinates is made. More precisely, the convergence rate for random initial

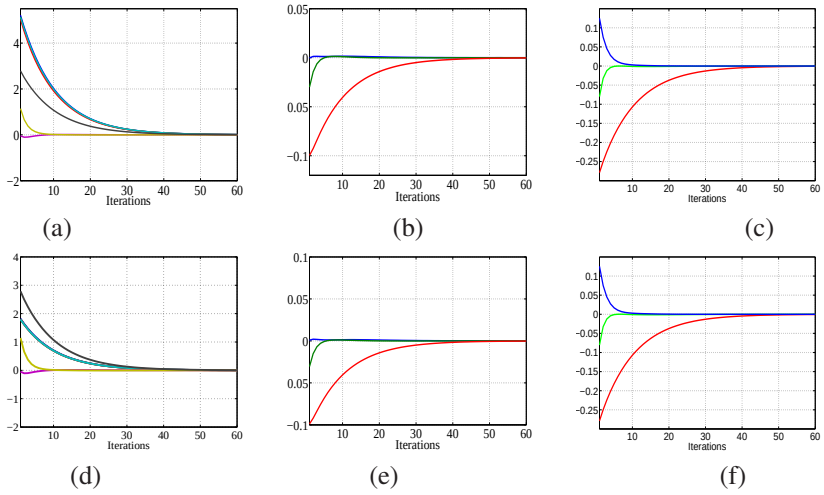


Fig. 8. Up: Results for large general motion using s_Δ (a) feature errors, (b) translational velocities(meters/s), (c) rotational velocities (rad/s)), down: results for large general motion using I_t (d) feature errors, (e) translational velocities (meters/s), (f) rotational velocities (rad/s))

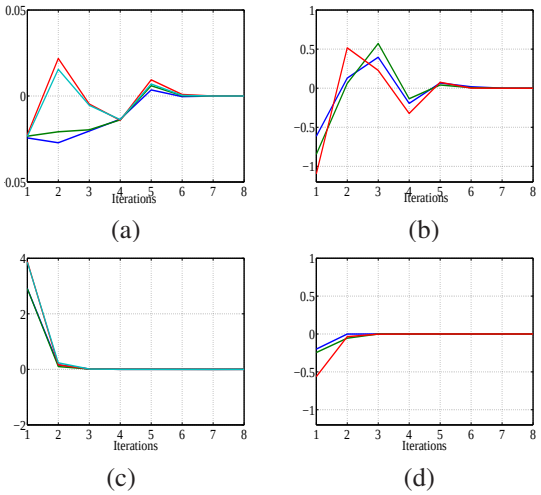


Fig. 7. Results obtained using Δ: (a) features errors, (b) velocities (m/s); obtained results using s_Δ: (c) features errors, (d) velocities (m/s)

and desired poses are computed using the different features in the case of perfect data as well as in the case when errors on camera calibration and noisy image points occur. The following setup has been used:

- Two objects composed respectively of four points forming a square (40) and six points generated randomly (41) have been considered (refer respectively to Figs 9.(a) and 9.(b)):

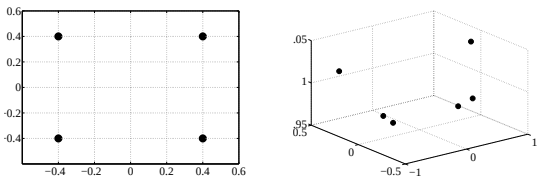
$$X_1 = \begin{bmatrix} -0.4 & 0.4 & -0.4 & 0.4 \\ -0.4 & -0.4 & 0.4 & 0.4 \\ 1. & 1. & 1. & 1 \end{bmatrix}. \qquad (40)$$

$$X_2 = \begin{bmatrix} -0.31 & 0.68 & -0.00 & -0.69 & 0.42 & -0.09 \\ 0.07 & -0.09 & -0.41 & 0.40 & -0.28 & 0.32 \\ 0.96 & 1.04 & 0.99 & 1.01 & 0.99 & 0.95 \end{bmatrix}. \qquad (41)$$

- A conventional camera model with focal $F = 800$ and principal point coordinates $u = v = 400$ pixels has been used to compute the image points coordinates.
- The interaction matrix corresponding to the current position is used in the control law (4) to compute the camera displacement (i.e $\widehat{\mathbf{L}}_\mathbf{s} = \mathbf{L}_\mathbf{s}$) and the scalar λ has been set to 0.1.
- The initial and the desired camera poses have been generated randomly as follow:
  - 600 random translational motions $\mathbf{t} = (1.5\,\sigma_1,\ 1.5\,\sigma_2,\ 1.7\sigma_3)$ are firstly applied to the point coordinates defined in the object frame, where σ_1 and σ_2 are random numbers chosen from a normal distribution with mean zero and standard deviation one, σ_3 is a random number chosen from a uniform distribution on the interval $[0.0\ 1.0]$.
  - The rotational motion is chosen such that the points coordinates belongs to the image limits $[1\ \ 800;\ 1\ \ 800]$. Further, the rotational motion with respect to the optical axis can range randomly between $[0\ \ 2\pi]$.

The obtained results using the square are given in Table I. The first line of Table I gives the percentage of convergence for the case when perfect data is used. It can be noticed that for the considered desired and initial poses, s_Δ allows obtaining the highest convergence rate, followed by I_t and then the Cartesian points coordinates. The results obtained with errors on camera parameters (10% errors on focal length and 20 pixels error on principal point coordinates) are similar and show the superiority of the feature s_Δ. Furthermore, the convergence rate does not suffer from the errors on camera parameters and noise.

Let us now discuss the results obtained using the object composed by 6 non-coplanar points. Two cases have been considered for s_Δ: in the first case the feature vector is defined by the invariants s_Δ computed for all possible 10 triangles obtained by combining three different points, while in the second case only 4 selected triangles are used. The 4 triangles are selected such that their centers of gravity are not collinear and the farthest possible from each other to allow a good conditioning of the interaction matrix (see Fig. 10). The Table



(a) (b)

Fig. 9. (a) the square point coordinates, (b) object composed by 6 points generated randomly

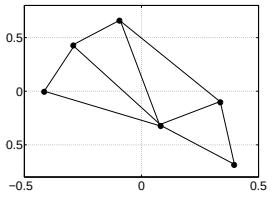


Fig. 10. Selected 4 triangles to compute the invariant s_Δ

II gives the convergence rate using respectively the features s_Δ (using 4 triangles), s_Δ (using all triangles), I_t (using all triangles) and finally the Cartesian point coordinates. The obtained results confirm those obtained using the square— the use of s_Δ allows obtaining a better convergence rate that the other features. Furthermore, the convergence rate when only 4 selected triangles are used to compute s_Δ increased significantly; using only 4 features to control the translational motions allowed avoiding local minima.

Finally, the statistical studies of convergence rate show that the majority of the cases when the camera does not converge to its desired position are mainly due to local minima. The cases of complete divergence happen usually because of errors on the used interaction matrix values.

In these last simulation results, we test the effect of the noise on image points coordinates on the computed velocities using s_Δ to control the translations. For this purpose, the object defined by (41) and an omnidirectional camera with focal $F = 500$, principal point coordinates $u = v = 300$ pixels and camera parameter $\xi = 2$ have been considered. The desired and the initial image points are given respectively on Figs 11.(a) and 11.(b). A generic motion combining the rotational motion $\boldsymbol{\theta}\mathbf{u} = (-27.88,\ 41.82,\ 69.70)$ degrees and the translational motion $\mathbf{t} = (-0.4,\ 0.2,\ .7)$meter is considered between the initial and the desired camera positions. Further, white noise with standard deviation equal to 0.5 pixels has been added to the image points coordinates. The obtained behaviors for the feature errors, the rotation velocities and the translational ones are given respectively on Figs 11.(c), 11.(d) and 11.(e). From these figures, it can be noticed that the computed velocities are once again satisfactory and robust to the considered noise on image points.

D. Experimental results using a conventional and a fisheye cameras

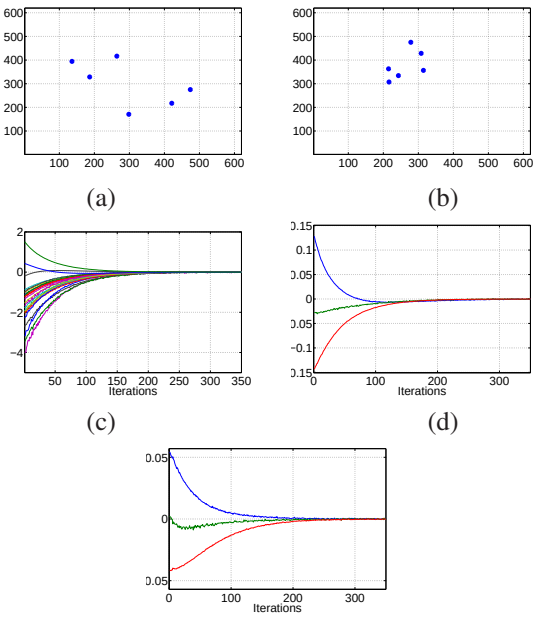
In the following, a series of experiments using conventional and fisheye cameras are presented. The parameters of the

Features	s_Δ	I_t	Point coordinates
Perfect data	100%	74.4%	62.1%
Noisy	99.83%	73.83%	62.33%

TABLE I
CONVERGENCE RATE USING THE SQUARE

s_Δ (4 triangles)	s_Δ (all triangles)	I_t (all triangles)	Point coordinates
99.8%	85.6%	86%	78.8%

TABLE II
CONVERGENCE RATE USING THE OBJECT COMPOSED OF 6 NON-COPLANAR POINTS



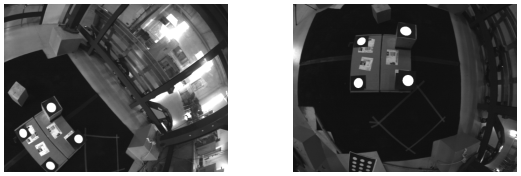
(a) (b) (c) (d) (e)

Fig. 11. Effect on noisy data on the computed velocities: (a) desired image points, (b) initial image points, (c) feature errors, (d) rotational velocities (rad/s), (e) translational velocities (meters/s)

classical perspective camera used in these experiments are: ($\xi = 0$, principal point coordinates $[u_x = 534.8,\ u_u = 353.4]$ pixels, the focal $[F_x = 960, F_y = 959]$ pixels). Those of the fisheye camera used in these experiments are: ($\xi = 1.71$, principal point coordinates $[u_x = 315.61,\ u_u = 243.05]$, the focal $[F_x = 722.91, F_y = 721.65]$). Two different objects composed respectively by 4 coplanar points and 4 non-coplanar points will be considered. The calibration of the two cameras was performed using the toolbox provided by Mei [20].

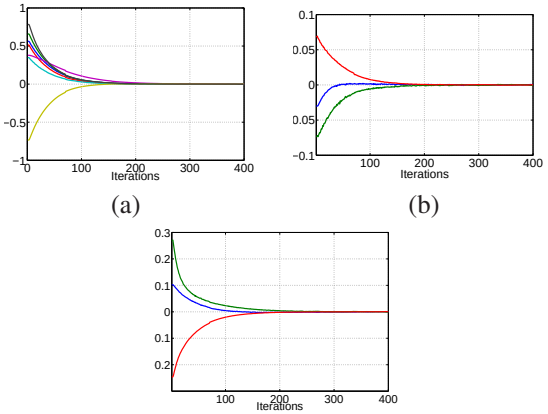
1) Results using a set of 4 non-coplanar points: In this part, an experimental result using fisheye camera and generic motion involving a large rotational and translational motions is presented. Both the behaviors of the control law using s_Δ and I_t are tested. The interaction matrix computed for the current camera position is used in the control law. The images corresponding to the camera initial and desired poses are given on Fig 12. The obtained results using s_Δ are given on Figure

13 and those using I_t are given on Fig. 14. The plots show that the error in feature value as well as the velocities converge to 0 in very satisfactory ways using s_Δ and I_t.



(a) (b)

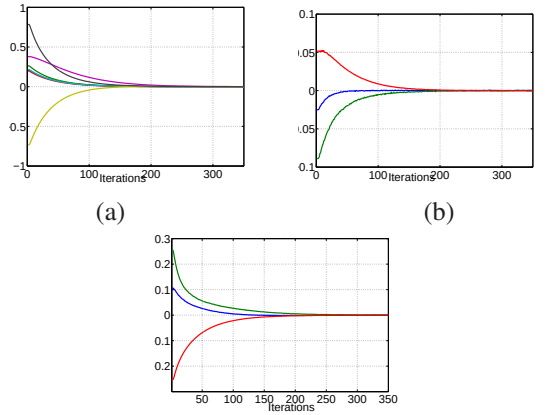
Fig. 12. Results using fisheye camera and a non-coplanar set of 4 points: (a) initial image, (b) desired image



(a) (b)

(c)

Fig. 13. Results using fisheye camera and a non-coplanar set of 4 points(generic motion using s_Δ): (a) feature errors, (b) translational velocities (meters/s), (c) rotational velocities (rad/s)



(a) (b)

(c)

Fig. 14. Results using fisheye camera and a non-coplanar set of 4 points (generic motion using I_t): (a) feature errors, (b) translational velocities (meters/s), (c) rotational velocities (degrees/s)

2) Results using a set of 4 *coplanar points:* In this part, experiments using the feature s_Δ to control the translational motions are presented.

a) Results using a perspective camera: In the following experiment, the interaction matrices are computed using the current values of the points in the image and constant approximated desired point depths. The first experiment involves pure rotational motion around the camera optical axis (80 degrees). The desired and the initial images are given respectively in Figs 15 and 16.(a). Four combinations of triangles obtained from the points defining the target are used to control translational motion. The results are shown in Figs 16.(b), 16.(c) and 16.(d). Fig. 16.(b) shows that a nice decrease of feature error is obtained. Furthermore, since the considered translational motion is null, the translational velocity computed using the invariants to rotations are almost null (refer to Fig. 16.(b)). The small translational velocities are due to the weak calibration of the camera. Fig.16.(d)) shows good behavior for rotational motions as well.

The second experiment using a perspective camera involves a complex motion between the initial and the desired camera positions and the corresponding images are shown in Fig. 17.(a) and Fig 15.(a) respectively. From Fig. 17.(b) we notice that the feature error behaves in a very satisfactory way despite the errors in camera calibration and points depth (the point depths are not computed at each iteration). Satisfactory behaviors are also obtained for translational and rotational velocities (see Figs 17.(c) and 17.(d)). Indeed, nice decreases of the feature errors as well as for the velocities are obtained.

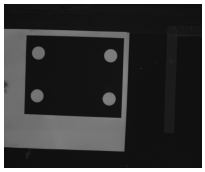
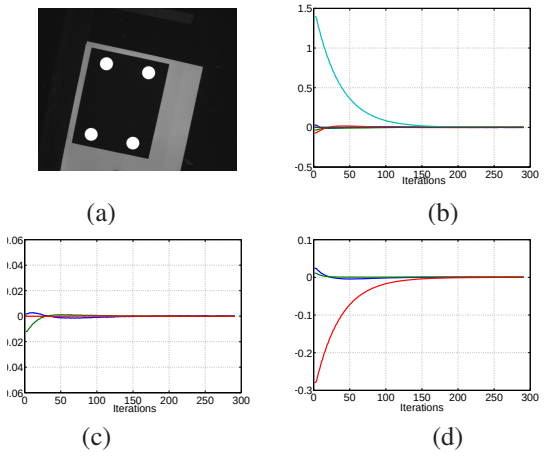


Fig. 15. Desired image using a conventional camera



(a) (b)

(c) (d)

Fig. 16. Results for pure rotation motion ($80dg$) using conventional camera: a) initial image, b) feature errors, c)translational velocities(meters/s), d) rotational velocities (rad/s)

b) Results using a fisheye camera: In this experiment, a generic displacement involving translational and rotational motion has been considered. The images corresponding to the initial and the desired camera positions are given respectively on Figs. 18.(a) and 18.(b). The displacement to perform is

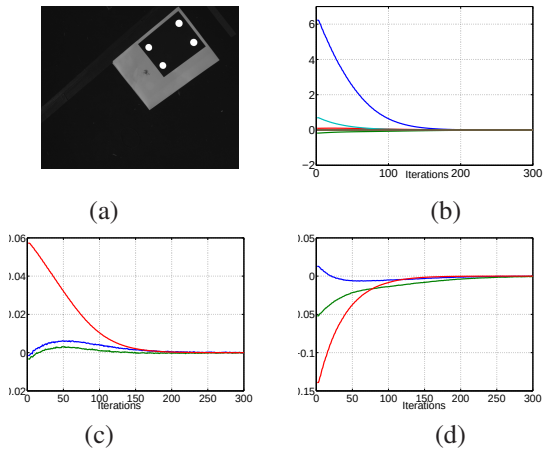


Fig. 17. Results for complex motion using conventional camera: a) initial image, b) feature errors, c)translational velocities(meters/s), d) rotational velocities (rad/s)

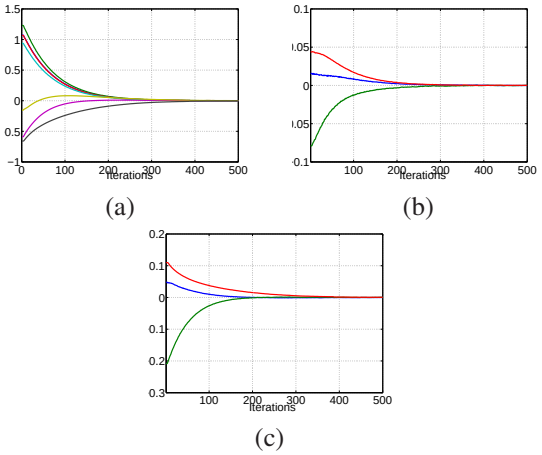


Fig. 19. Results for 4 coplanar points (generic motion using s_Δ): (a) feature errors, (b) translational velocities (meters/s), (c) rotational velocities (rad/s)

very large and the camera desired orientation with respect to the object plane is approximately equal to 60 degrees. In the control law, the interaction matrix computed for the current camera position is used. The obtained results are given on Fig. 19. From the plots, it can be noticed that very satisfactory decrease of the error on features as well as for the velocities is obtained. This shows that the orientation of the camera in its desired position does not impact the behavior of the proposed control scheme. This result was expected since a rotation motion of the camera only introduces a rotation of the interaction matrix related to the invariant to the rotations.

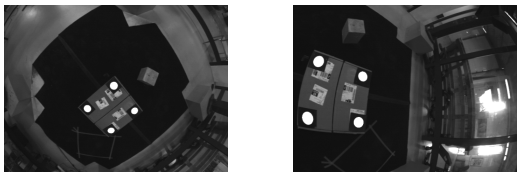


Fig. 18. Results for 4 coplanar points (generic motion using s_Δ): (a) initial image, (b) desired image

E. Discussion

We have shown that two different features can be obtained from a set of points to control the translational DOFs: the features s_Δ obtained from the surfaces of the projection of triangles onto a unit sphere and the polynomials invariant to rotations I_t. The results we have obtained show that the two invariants provide good decoupling properties.

This raises the question "*which is the best feature to use?*". The experimental results and more precisely the convergence rate using two different objects composed respectively by 4 and 6 points have shown the superiority of the control using s_Δ. However, in the case when the target comprises many points the number of possible combinations of three points increases combinatorially. This means that the size of the features vector increases and reaching local minima becomes possible. In order to avoid those local minima we can consider using only 4 the *best* triangles, for instance those triangles that result in the best conditioning of the interaction matrix for the desired position. However since not all points would be used for servoing the robustness to noise might decrease. In order to improve the robustness to noise, all the possible combinations of points can be used near the desired position.

Alternatively the invariant polynomial could be computed from more than three points. Therefore if the set of points is divided into only four different subsets we can obtain an almost minimal representation (four features to control the three translational DOFs).

The projection surface has also the advantage to be suitable for points but also for closed contours. The surface of the object's projection onto a sphere is simply the moment of order 0 and it can be computed using the general formula:

$$m_{s_{i,j,k}} = \iint_{region} x_s^i y_s^j z_s^k ds \tag{42}$$

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. As for the surfaces obtained by projection of triangles, it is also possible to show that the feature $\frac{1}{\sqrt{m_{000}}}$ has the same properties as s_Δ and I_t with respect to object depth variation. The decoupled control proposed for objects defined by a set of points can thus be extended straightforwardly to the case where several matched planar contours are available in the scene. More precisely, three planar contours at least are required to control the three translational degrees of freedom.

In order to control the rotational degrees of freedom, similar features to those proposed for the case of objects defined by a set of point can also be used. Indeed, the center of gravity coordinates defined as $\mathbf{x}_{\mathbf{s_g}} = \left(\frac{m_{100}}{m_{000}}, \frac{m_{010}}{m_{000}}, \frac{m_{001}}{m_{000}}\right)$ can also be computed using the moment definition (42). Finally, in order to control the rotation around the optical axis, the object orientation in the image ($\theta = \frac{1}{2} arctan(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}})$) can be used as in [26]. The interaction matrix related to region-based moments computed from the projection onto sphere has been already computed in [25]. The analytical formulas of the

features as well as their related interaction matrices required for the control are available. The extension of the image-based control obtained for a set of points to the case of multiple matched planar-contours is then possible.

V. Conclusions and future work

In this paper a generic decoupled image-based control using the projection onto the unit sphere was proposed. Invariants to rotation have been used to control translational motion. The proposed decoupled control is valid for all cameras obeying the unified camera model. Further, it is valid for objects defined by at least three planar closed contours or by a set of at least four points. Importantly, the proposed features result in an interaction matrix whose elements are only weakly dependent on the depth of object points and camera position. Finally, the controller has been experimentally validated and results presented using two kinds of camera: conventional and fisheye. The results show very satisfactory behavior in both 3D space and the image. Future work will be devoted to extending these results to the pose estimation problem.

References

[1] S. Baker and S. Nayar. A theory of catadioptric image formation. *Int. Journal of Computer Vision*, 35(2):175–196, November 1999.

[2] J. Barreto and H. Araujo. Issues on the geometry of central catadioptric image formation. In *Computer Vision and Pattern Recognition, 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–422–II–427 vol.2, 2001.

[3] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. . Hager, and A. Morse, editors, *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer-Verlag, 1998.

[4] G. Chesi. Visual servoing path-planning via homogeneous forms and lmi optimizations. *IEEE Transactions on Robotics*, 25(5):281–291, 2009.

[5] G. Chesi and A. Vicino. Visual servoing for large camera displacements. *IEEE Transactions on Robotics*, 20(4):724 –735, 2004.

[6] 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, August 2001.

[7] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8:313–326, June 1992.

[8] J. Feddema and O. Mitchell. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5):691–700, Oct 1989.

[9] D. Fioravanti, B. Allotta, and A. Rindi. Image based visual servoing for robot positioning tasks. *Meccanica*, 43(3):291–305, June 2008.

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

[11] H. Hadj-Abdelkader, Y. Mezouar, P. Martinet, and F. Chaumette. Catadioptric visual servoing from 3d straight lines. *IEEE Transactions on Robotics*, 24(3):652–665, June 2008.

[12] E. Hall. *Computer Image Processing and Recognition*. Academic Press, 1979.

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

[14] M. Iwatsuki and N. Okiyama. A new formulation of visual servoing based on cylindrical coordinates system with shiftable origin. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 354–359, Lausanne, Switzerland, October 2002.

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

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

[17] 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.

[18] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1843– 1848, New Orleans, Louisiana, April 2004.

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

[20] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE Int. Conf. on Robotics and Automation*, pages 3945–3950, April 2007.

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

[22] P. Rives and J. Azinheira. Linear structures following by an airship using vanishing points and horizon line in a visual servoing scheme. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, pages 255–260, New Orleans, Louisiana, April 2004.

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

[24] T. Svoboda and T. Pajdla. Epipolar geometry for central catadioptric cameras. *Int. Journal on Computer Vision*, 49(1):23–37, August 2002.

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

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

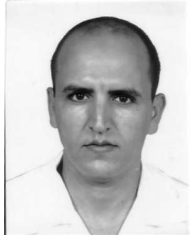
[27] O. Tahri, F. Chaumette, and Y. Mezouar. New decoupled visual servoing scheme based on invariants from projection onto a sphere. In *IEEE Int. Conf. on Robotics and Automation*, pages 3238–3243, May 2008.

[28] O. Tahri and Y. Mezouar. On visual servoing based on efficient second order minimization. *Robotics and Autonomous Systems*, 2010. To apear.

[29] O. Tahri, Y. Mezouar, F. Chaumette, and P. Corke. Generic decoupled image-based visual servoing for cameras obeying the unified projection model. In *IEEE Int. Conf. on Robotics and Automation, ICRA'09*, pages 1116–1121, Kobe, Japan, May 2009.

[30] R. Tatsambon Fomena and F. Chaumette. Visual servoing from spheres using a spherical projection model. In *IEEE International Conf. on Robotics and Automation*, pages 2080–2085, April 2007.

[31] 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.



Omar Tahri was born in Fez, Morocco, in 1976. He got his Masters in photonics, images and system control from the Louis Pasteur University, Strasbourg, France, in 2000 and received his Ph.D degree in computer science from the University of Rennes, France, in March 2004. Since 2008, he is working as a researcher with the computer vision lab (http://labvis.isr.uc.pt/new/index.html) of the institute for Systems and Robotics, Coimbra, Portugal (http://www.isr.uc.pt/home.php). His research interests include robotics and computer vision, especially visual servoing.

Youcef Mezouar received the Ph.D degree in computer science from the Université de Rennes 1, Rennes, France in 2001 and the "Habilitation à diriger les recherches" degree in 2009 from the Université Blaise Pascal, Clermont-Ferrand, France in 2009. He spent one year as Postdoctoral Associates in the Robotics Lab of the Computer Science Department at Columbia University, New York, USA. He joined the Robotics and Vision Group of LASMEA-CNRS, Clermont-Ferrand, France in 2002. He currently co-leads the GRAVIR group (`http://wwwlasmea.univ-bpclermont.fr`) and the ROSACE team (`http://wwwlasmea.univ-bpclermont.fr/rosace`). His research interests include automatics, robotics and computer vision, especially visual servoing and mobile robots navigation.

François Chaumette was graduated from École Nationale Supérieure de Mécanique, Nantes, France, in 1987. He received the Ph.D. degree in computer science from the University of Rennes, France, in 1990. Since 1990, he has been with INRIA in Rennes where he is now "Directeur de Recherches" and head of the Lagadic group (`http://www.irisa.fr/lagadic`). His research interests include robotics and computer vision, especially visual servoing and active perception.

Dr. Chaumette received the AFCET/CNRS Prize for the best French thesis in automatic control in 1991. He also received with Ezio Malis the 2002 King-Sun Fu Memorial Best IEEE Transactions on Robotics and Automation Paper Award. He has been Associate Editor of the IEEE Transactions on Robotics from 2001 to 2005 and is now in the Editorial Board of the Int. Journal of Robotics Research.

Peter Corke is a Professor with the School of Engineering Systems at the Queensland University of Technology. Previously he was founding, Research Director of the Autonomous Systems Laboratory in the CSIRO ICT Centre. His research activities span machine vision, vision-based robot control, field robotics and sensor networks. He received his PhD from the University of Melbourne. He is the editor-in-chief of the IEEE Robotics&Automation magazine, a member of the editorial board of the International Journal of Robotics Research and a founding editor of the Journal of Field Robotics.