Distance-Based and Orientation-Based Visual Servoing From Three Points

Romeo Tatsambon Fomena, *Member, IEEE*, Omar Tahri, and François Chaumette, *Senior Member, IEEE*

***Abstract*—This paper is concerned with the use of a spherical-projection model for visual servoing from three points. We propose a new set of six features to control a 6-degree-of-freedom (DOF) robotic system with good decoupling properties. The first part of the set consists of three invariants to camera rotations. These invariants are built using the Cartesian distances between the spherical projections of the three points. The second part of the set corresponds to the angle-axis representation of a rotation matrix measured from the image of two points. Regarding the theoretical comparison with the classical perspective coordinates of points, the new set does not present more singularities. In addition, using the new set inside its nonsingular domain, a classical control law is proven to be optimal for pure rotational motions. The theoretical results and the robustness to points range errors of the new control scheme are validated through simulations and experiments on a 6-DOF robot arm.**

***Index Terms*—Three points, spherical projection, visual servoing.**

I. Introduction

VISUAL servoing consists of using feedback information provided by a vision sensor to control the motion of a dynamic system [1], [2]. A vision sensor provides a large spectrum of potential visual features. However, if no planning strategy is developed, the use of some visual features in the control scheme may lead to stability problems in the case where the displacement that the robot has to achieve is very large [3]. This justifies the need to design ideal visual features for visual servoing. By ideal, satisfaction of the following criteria is meant: local and, as far as possible, global stability of the system, robustness to calibration and modeling errors, nonsingularity, local-mimima avoidance, satisfactory trajectory of the system and of the features in the image, and, finally, maximal decoupling and linear link (which is the ultimate goal) between the visual features and the degrees of freedom (DOFs) taken into account.

Points are the most simple features that can be extracted from an image, both from a geometrical and an image processing points of view. This explains why most of the visual-servoing applications are based on image of points, for instance, visual homing [4], autonomous navigation of mobile robots [5], [6], and autonomous stabilization of aerial vehicles [7], [8].

The image of points is at the center of lots of works devoted to approach an ideal-system behavior using a 3-D, a hybrid (i.e., 3-D and 2-D), or a 2-D visual-servoing method. In a 3-D visual servoing, the image of points and the knowledge of the object geometry serve to compute the relative pose of the object with respect to (w.r.t.) the camera, which is then used to elaborate visual features in 3-D Cartesian space [9], [10]. Estimating 3-D data or pose can be useful to ensure a global convergence when image noise and calibration errors are not present [11]. However, small image noise could lead to large errors in the pose estimation [3]. In hybrid visual servoing, which combines both 2-D and 3-D data, the 3-D data are generally obtained via the estimation of an homography matrix between the current and the desired images. This solution has been exploited in 2-1/2-D visual servoing, where a partially decoupled control scheme with no singularity in the whole task space and tolerating a coarsely calibrated system has been designed [12]. Using a coplanar set of points as a target, a decoupled homography-based visual servoing scheme that builds on the knowledge of the 3-D distance between two points has been proposed in [13]. However, hybrid visual servoing is more sensitive to image noise (like 3-D visual servoing) than 2-D visual servoing, which uses directly features extracted in the image as control inputs. In 2-D visual servoing, simple perspective coordinates of image of points can be used to control a system [14]; however, the resulting control law is highly coupled and has a limited convergence domain [3]. Possible decoupling solutions exist for the motion along and around the optical axis, for instance, the cylindrical-coordinate approach [15] or the partitioned approach [16]. Another appealing solution to the decoupling issue is the use of moment-invariant theory. Using this approach, a combination of 2-D moments has been proposed for planar objects that are defined by points cloud [17]. Lately, spherical moments invariant to camera rotations, such as the area of a surface, have been used to design a partially decoupled control scheme for volumetric and planar objects defined by more than three points [18]. Finally, in the case of a very particular target composed of a sphere marked with two points on its surface, a spherical-projection model has been used to propose a set of features providing a partially decoupled control system [19].

Inspired by the aforementioned work [18], this paper exploits the geometric properties of the spherical projection of three points and proposes a new set of six features. An early version of this work appeared in [20]. This paper extends that work by improving the theoretical developments and by validating the approach on a real robotic system.

Manuscript received May 26, 2010; accepted December 29, 2010. Date of publication February 28, 2011; date of current version April 7, 2011. This paper was recommended for publication by Associate Editor C. C. Cheah and Editor G. Oriolo upon evaluation of the reviewers' comments. This paper was presented in part at the International Conference on Robotics and Automation, Anchorage, AK, May 2010.

R. Tatsambon Fomena is with Université de Rennes 1, IRISA, 35 042 Rennes-Cedex, France (e-mail: rtatsamb@irisa.fr).

O. Tahri is with the ISR, University of Coimbra, Polo II PT-3030-290 Coimbra, Portugal (e-mail: omartahri@isr.uc.pt).

F. Chaumette is with INRIA Rennes-Bretagne Atlantique and IRISA, 35 042 Rennes-Cedex, France (e-mail: chaumett@inria.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2011.2104431

Even if it is well known that the same image of three points corresponds to four different camera poses [21], it is possible to control a 6-DOF robot using only three points in a neighborhood of the desired camera pose. The originality presented in this paper in comparison with [18] is that the decoupling is obtained by three features invariant to rotations that are the Cartesian distances between the spherical projections of the three points. In addition, the three other features, which are designed to control rotation motions, are built using the angle-axis representation of a rotation matrix measured from the image of two points. The interaction matrix associated with the proposed angle-axis representation is valid for a general configuration between points, including the specific configuration between points proposed in [22], where the target is a sphere marked with a tangent vector to a point on its surface.

Feature modeling and associated interaction matrices are detailed in Section II. In Section III, we compare our approach with the classical perspective coordinates of three points, which has been proven to present a singularity domain defined by a cylinder [23]. In addition, due to the coupling in their interaction matrix, the perspective coordinates of points are not suited for pure camera rotations [3]. A key contribution of this paper is the formal demonstration that our approach does not present more singularities and is optimal for pure camera rotations compared with the classical perspective coordinates of three points and the use of the coordinates of the center of gravity in [18]. Concerning the singularity issue, the demonstration given in this paper is simple and easy compared with the demonstration using the perspective coordinates given in [23]. The proposed control scheme depends on the ranges of points that are generally unknown in practice, as in the case of the classical perspective projection. In Section III, we also analyze the robustness of the control w.r.t. error on depth-value estimation. Finally, simulation results using the true depth values are presented in Section IV to validate the proposed approach in ideal conditions. In practice, the depth values could be estimated using structure from motion methods [24]. However, in order to show the robustness w.r.t. estimation errors, a very coarse approximation of the depth values, i.e., the desired values, is rather used in experiments on a 6-DOF eye-in-hand robotic system.

II. Feature Modeling

In this section, we design a set $\mathbf{s_{sp}} = (\mathbf{s}_t, \boldsymbol{\xi})$ of six features to control the image of three points using a spherical-projection model. The set of features $\mathbf{s}_t$ is invariant to camera rotations, while the set $\boldsymbol{\xi}$ is the angle-axis representation of a rotation matrix defined from the image of two points.

Let $\mathcal{S}_{(\mathcal{C},1)}$ be the unit sphere of projection center in $\mathcal{C}$; $\mathcal{F}_c = (\mathcal{C}, \mathbf{x}, \mathbf{y}, \mathbf{z})$ be the frame attached to the camera unique projection center $\mathcal{C}$; $\mathcal{P}_0$, $\mathcal{P}_1$, and $\mathcal{P}_2$ be a set of three points; and $\mathbf{P}_i = (P_{ix}, P_{iy}, P_{iz})$ be the vector coordinates of $\mathcal{P}_i$ in $\mathcal{F}_c$.

We first recall that the interaction matrix $\mathbf{L_f}$ related to a set of features $\mathbf{f} \in \mathbb{R}^n$ is defined such that $\dot{\mathbf{f}} = \mathbf{L_f}\mathbf{v}_c$, where $\mathbf{v}_c = (\boldsymbol{v}, \boldsymbol{\omega}) \in \mathfrak{se}(\mathbf{3})$ is the instantaneous camera velocity [25], $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are, respectively, the translational and the rotational velocities of the camera, and $\mathfrak{se}(\mathbf{3}) \simeq \mathbb{R}^3 \times \mathbb{R}^3$ is the Lie algebra of the Lie group of displacements $\mathbf{SE}(\mathbf{3})$.

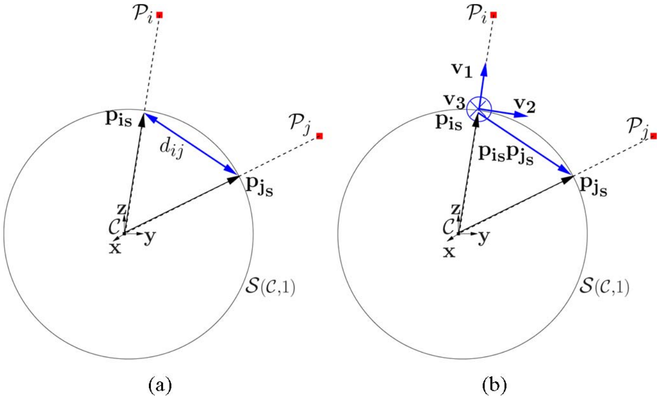


Fig. 1. Spherical projection of two points. (a) Distance between the spherical images of the two points and (b) components of the rotation matrix.

A. Set $\mathbf{s}_t$ Invariant to Camera Rotations

Let $\mathbf{p_{i_s}}$ be the spherical projection of points $\mathcal{P}_i$. We recall that the spherical projection of a point $\mathcal{P}_i$ is defined by $\mathbf{p_{i_s}} = \mathbf{P}_i/\|\mathbf{P}_i\|$. Note that $\mathbf{p_{i_s}}$ can easily be measured from a perspective image using formula $\mathbf{p_{i_s}} = \mathbf{p_{i_p}}/\|\mathbf{p_{i_p}}\|$ with $\mathbf{p_{i_p}} = (P_{ix}/P_{iz}, P_{iy}/P_{iz})$ or from any omnidirectional image with a unique projection center [26]. Let d_{ij} be the Cartesian distance between $\mathbf{p_{i_s}}$ and $\mathbf{p_{j_s}}$ [see Fig. 1(a)]. The analytical expression of d_{ij} is given by

$$d_{ij} = \|\mathbf{p_{i_s}} - \mathbf{p_{j_s}}\| = \sqrt{2 - 2\mathbf{p_{is}^\top p_{js}}}. \tag{1}$$

Since a camera frame rotation preserves Cartesian distances, it is straightforward that feature d_{ij} is invariant to camera rotation motions, as demonstrated below. From (1), the time variation of d_{ij} is given by

$$\dot{d}_{ij} = -\frac{1}{d_{ij}}\left(\mathbf{p_{is}^\top \dot{p}_{js}} + \mathbf{p_{js}^\top \dot{p}_{is}}\right). \tag{2}$$

The time variation of $\mathbf{p_{k_s}}$ is given by [7]

$$\dot{\mathbf{p}}_{\mathbf{k_s}} = -\frac{1}{\|\mathbf{P}_k\|}\Gamma_{\mathbf{p_{k_s}}}\boldsymbol{v} + [\mathbf{p_{k_s}}]_\times\boldsymbol{\omega} \tag{3}$$

where $\Gamma_{\mathbf{p_{k_s}}} = [\mathbf{p_{k_s}}]_\times^2 = \mathbf{I}_3 - \mathbf{p_{k_s}}\mathbf{p_{k_s}}^\top$, and $[\mathbf{p_{k_s}}]_\times$ is the antisymmetric matrix related to vector $\mathbf{p_{k_s}}$. Note that the range $\|\mathbf{P}_k\|$, which is generally unknown in practice, appears in the translation part of $\dot{\mathbf{p}}_{\mathbf{k_s}}$ similarly as the depth P_{kz} appears in the well-known interaction matrix of the perspective image of a point.

Substituting (3) into (2), we immediately obtain the expression of the interaction matrix related to d_{ij}

$$\mathbf{L}_{d_{ij}} = \left[\frac{1}{d_{ij}}\left(\frac{1}{\|\mathbf{P}_j\|}\mathbf{p_{is}^\top}\Gamma_{\mathbf{p_{js}}} + \frac{1}{\|\mathbf{P}_i\|}\mathbf{p_{js}^\top}\Gamma_{\mathbf{p_{is}}}\right) \quad \mathbf{0}_{\{1\times 3\}}\right]. \tag{4}$$

The bloc $\mathbf{0}_{\{1\times 3\}}$ on the rotation component of the interaction matrix $\mathbf{L}_{d_{ij}}$ clearly shows the invariance property. This block has also been highlighted in [18] when using invariants of spherical moments built from the image of at least four points, whereas in this paper, we use only three points.

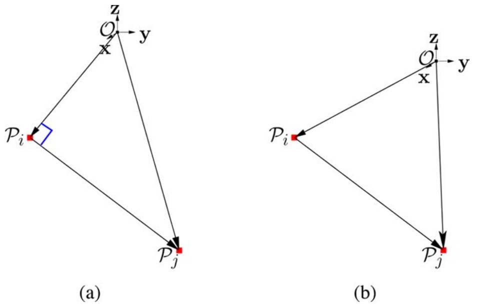


Fig. 2. Configuration of the two points in object frame $\mathcal{F}_O = (\mathcal{O}, \mathbf{x}, \mathbf{y}, \mathbf{z})$. (a) Specific case presented in [22] and (b) general case.

The set $\mathbf{s}_t = (d_{01}, d_{02}, d_{12})$ is thus formed by selecting the three distances between the spherical projection of the three points. Its interaction matrix is given by

$$\mathbf{L}_{\mathbf{s}_t} = [\,\mathbf{L}_v \quad \mathbf{0}\,] \tag{5}$$

with

$$\mathbf{L}_v = \begin{bmatrix} \frac{1}{d_{01}}\left(\frac{1}{\|\mathbf{P}_1\|}\mathbf{p_{0s}}^\top\Gamma_{\mathbf{p_{1s}}} + \frac{1}{\|\mathbf{P}_0\|}\mathbf{p_{1s}}^\top\Gamma_{\mathbf{p_{0s}}}\right) \\ \frac{1}{d_{02}}\left(\frac{1}{\|\mathbf{P}_2\|}\mathbf{p_{0s}}^\top\Gamma_{\mathbf{p_{2s}}} + \frac{1}{\|\mathbf{P}_0\|}\mathbf{p_{2s}}^\top\Gamma_{\mathbf{p_{0s}}}\right) \\ \frac{1}{d_{12}}\left(\frac{1}{\|\mathbf{P}_2\|}\mathbf{p_{1s}}^\top\Gamma_{\mathbf{p_{2s}}} + \frac{1}{\|\mathbf{P}_1\|}\mathbf{p_{2s}}^\top\Gamma_{\mathbf{p_{1s}}}\right) \end{bmatrix}.$$

B. Angle-Axis $\boldsymbol{\xi}$ Representation of a Rotation Matrix

From the spherical projections $\mathbf{p_{is}}$ and $\mathbf{p_{js}}$, it is possible to determine a set of three features $\boldsymbol{\xi}$ such that its interaction matrix has the form

$$\mathbf{L}_{\boldsymbol{\xi}} = [\,-\mathbf{L}_\omega(\theta_\xi, \mathbf{u}_\xi)\mathbf{L}_{\omega,v} \quad \mathbf{L}_\omega(\theta_\xi, \mathbf{u}_\xi)\,] \tag{6}$$

where $\mathbf{L}_{\omega,v}$ and $\mathbf{L}_\omega(\theta_\xi, \mathbf{u}_\xi)$ will be defined later. The set $\boldsymbol{\xi}$ is based on two ideas. The first idea, as proposed in [12], concerns the $\theta\mathbf{u}$ (where θ is the angle of rotation, and $\mathbf{u}$ is the unitary axis of rotation) representation of the relative rotation between the current and desired camera frames. The second idea, as proposed in [22], is about visual servoing from a sphere marked with a tangent vector where the configuration between the two points $\mathcal{P}_i$ and $\mathcal{P}_j$ in the object frame $\mathcal{F}_o = (\mathcal{O}, \mathbf{x}, \mathbf{y}, \mathbf{z})$ is such that $\mathcal{O}\mathcal{P}_i \perp \mathcal{P}_i\mathcal{P}_j$ [see Fig. 2(a)]. As a major theoretical improvement of [20], the set $\boldsymbol{\xi}$ is defined as the $\theta\mathbf{u}$ representation of the rotation matrix ${}^{c}\mathbf{R}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*}$, where ${}^{c*}\mathbf{R}_{p^*_{is}}$ is the desired value of the matrix ${}^{c}\mathbf{R}_{p_{is}} = [\mathbf{v_1} \quad \mathbf{v_2} \quad \mathbf{v_3}]$ defined as follows [see Fig. 1(b)]:

$$\begin{cases} \mathbf{v_1} = \mathbf{p_{is}} \\ \mathbf{v_2} = \dfrac{\Gamma_{\mathbf{p_{is}}}(\mathbf{p_{js}} - \mathbf{p_{is}})}{\|\Gamma_{\mathbf{p_{is}}}(\mathbf{p_{js}} - \mathbf{p_{is}})\|} \\ \mathbf{v_3} = \mathbf{v_1} \times \mathbf{v_2} \end{cases} \tag{7}$$

where $\mathbf{v_2}$ is the projection of vector $\mathbf{p_{js}} - \mathbf{p_{is}}$ on the orthogonal plane to $\mathbf{p_{is}}$. From (7), it is clear that ${}^{c}\mathbf{R}_{p_{is}}$ is not defined only if $\mathcal{C}$, $\mathcal{P}_i$, and $\mathcal{P}_j$ are aligned.

The tangent space $T_{{}^{c}\mathbf{R}_{p_{is}}}\mathbf{SO(3)}$ to the element ${}^{c}\mathbf{R}_{p_{is}} \in \mathbf{SO(3)}$ is identified with $\mathfrak{so}(\mathbf{3}) \simeq \mathbb{R}^3$ via right translation by $({}^{c}\mathbf{R}_{p_{is}}, {}^{c}\dot{\mathbf{R}}_{p_{is}}) \to ({}^{c}\mathbf{R}_{p_{is}}, \boldsymbol{\zeta})$, where $\boldsymbol{\zeta} \in \mathfrak{so}(\mathbf{3})$ is such that

$$[\boldsymbol{\zeta}]_\times = {}^{c}\dot{\mathbf{R}}_{p_{is}}{}^{p_{is}}\mathbf{R}_c. \tag{8}$$

In the specific configuration, where $\mathcal{O}\mathcal{P}_i \perp \mathcal{P}_i\mathcal{P}_j$, from (8), it has been shown in [22] that

$$\boldsymbol{\zeta} = \mathbf{L}_{\omega,v}\boldsymbol{v} - \boldsymbol{\omega} \tag{9}$$

where the expression of matrix $\mathbf{L}_{\omega,v}$ is given by

$$\mathbf{L}_{\omega,v} = \frac{1}{\|\mathbf{P}_i\|}(\delta_c\mathbf{v_1}\mathbf{v_3}^\top + \mathbf{v_2}\mathbf{v_3}^\top - \mathbf{v_3}\mathbf{v_2}^\top) \tag{10}$$

with $\delta_c = -\mathbf{r_3}^\top\mathbf{v_2}/\mathbf{r_3}^\top\mathbf{v_1}$, where $\mathbf{r_3} = \mathbf{P_i O}/\|\mathbf{P_i O}\|$, with $\mathbf{O}$ the vector coordinates of the center of object frame $\mathcal{O}$ in the camera frame $\mathcal{F}_c$.

In this paper, we propose to compute $\boldsymbol{\zeta}$ for a general configuration between the points, as shown in Fig. 2(b). With that generalization, matrix $\mathbf{L}_{\omega,v}$ is now given by (for detailed developments, see Appendix A)

$$\mathbf{L}_{\omega,v} = \frac{1}{\|\mathbf{P}_i\|}(\delta_p\mathbf{v_1}\mathbf{v_3}^\top + \mathbf{v_2}\mathbf{v_3}^\top - \mathbf{v_3}\mathbf{v_2}^\top) \tag{11}$$

with $\delta_p = \frac{(\mathbf{p_{is}}^\top\mathbf{p_{js}})\|\mathbf{P}_j\| - \|\mathbf{P}_i\|}{\|\mathbf{P}_j\|\|\Gamma_{\mathbf{p_{is}}}(\mathbf{p_{js}} - \mathbf{p_{is}})\|}$. In addition, note here that the ranges $\|\mathbf{P}_i\|$ and $\|\mathbf{P}_j\|$ appear in the translation part of the interaction matrix. We will see in Section III-B how to deal with this issue.

In the case of the specific configuration $\mathcal{O}\mathcal{P}_i \perp \mathcal{P}_i\mathcal{P}_j$ [see Fig. 2(a)], it is possible to show that the general expression (11) equals the specific expression (10). Indeed, plugging the expressions of $\mathbf{v_1}$ and $\mathbf{v_2}$ given in (7) into δ_c gives (after expansion) $\delta_c\|\Gamma_{\mathbf{p_{is}}}(\mathbf{p_{js}} - \mathbf{p_{is}})\| = (\mathbf{p_{is}}^\top\mathbf{p_{js}}) - \mathbf{r_3}^\top\mathbf{p_{js}}/\mathbf{r_3}^\top\mathbf{p_{is}}$. By exploiting $\mathcal{O}\mathcal{P}_i \perp \mathcal{P}_i\mathcal{P}_j$, it is easy to show that $\mathbf{r_3}^\top\mathbf{p_{js}}/\mathbf{r_3}^\top\mathbf{p_{is}} = \|\mathbf{P}_i\|/\|\mathbf{P}_j\|$. Therefore, $\delta_c = \delta_p$.

Now, we use $\boldsymbol{\zeta}$ to develop the interaction matrix related to $\boldsymbol{\xi} = \theta\mathbf{u}({}^{c}\mathbf{R}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*})$. The expression of $\boldsymbol{\zeta}$ given in (8) can be rewritten as

$$[\boldsymbol{\zeta}]_\times = \frac{d({}^{c}\mathbf{R}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*})}{dt}({}^{c}\mathbf{R}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*})^\top \tag{12}$$

since

$$\begin{aligned} d({}^{c}\mathbf{R}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*})/dt &= {}^{c}\dot{\mathbf{R}}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*} + {}^{c}\mathbf{R}_{p_{is}}{}^{p^*_{is}}\dot{\mathbf{R}}_{c*} \\ &= {}^{c}\dot{\mathbf{R}}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*} \end{aligned}$$

because ${}^{p^*_{is}}\dot{\mathbf{R}}_{c*} = \mathbf{0}$. Plugging into (12) the Rodriguez formulation of ${}^{c}\mathbf{R}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*}$

$${}^{c}\mathbf{R}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*} = \mathbf{I_3} + \sin(\theta_\xi)[\mathbf{u}_\xi]_\times + (1 - \cos(\theta_\xi))[\mathbf{u}_\xi]^2_\times \tag{13}$$

where $\theta_\xi = \|\boldsymbol{\xi}\|$ and $\mathbf{u}_\xi = \boldsymbol{\xi}/\|\boldsymbol{\xi}\|$, we obtain after some developments (exactly as in [27])

$$\dot{\boldsymbol{\xi}} = -\mathbf{L}_\omega(\theta_\xi, \mathbf{u}_\xi)\boldsymbol{\zeta} \tag{14}$$

where

$$\mathbf{L}_\omega(\theta_\xi, \mathbf{u}_\xi) = -\mathbf{I} + \frac{\theta_\xi}{2}[\mathbf{u}_\xi]_\times - \left(1 - \frac{\text{sinc}(\theta_\xi)}{\text{sinc}^2(\theta_\xi/2)}\right)[\mathbf{u}_\xi]^2_\times \tag{15}$$

with $\mathrm{sinc}(x) = \sin(x)/x$. Finally, plugging the tangent space identification (9) into (14), we immediately obtain

$$\dot{\boldsymbol{\xi}} = -\mathbf{L}_{\omega}(\theta_{\xi}, \mathbf{u}_{\xi})\mathbf{L}_{\omega,\upsilon}\boldsymbol{\upsilon} + \mathbf{L}_{\omega}(\theta_{\xi}, \mathbf{u}_{\xi})\boldsymbol{\omega} \tag{16}$$

from which we deduce the interaction matrix of $\boldsymbol{\xi} = \theta\mathbf{u}({}^{c}\mathbf{R}_{p_{is}}{}^{p^*_{is}}\mathbf{R}_{c*})$ given in (6). Note that it would also be possible to use the $\theta\mathbf{u}$ representation of matrix ${}^{c*}\mathbf{R}_{p^*_{is}}{}^{p_{is}}\mathbf{R}_{c}$. In that case, by denoting $\boldsymbol{\xi}' = \theta_{\xi'}\mathbf{u}_{\xi'}$, we obtain

$$\mathbf{L}_{\omega}(\theta_{\xi'}, \mathbf{u}_{\xi'}) = \mathbf{I} + \frac{\theta_{\xi'}}{2}[\mathbf{u}_{\xi'}]_{\times} + \left(1 - \frac{\mathrm{sinc}(\theta_{\xi'})}{\mathrm{sinc}^2(\theta_{\xi'}/2)}\right)[\mathbf{u}_{\xi'}]^2_{\times}. \tag{17}$$

Parameters $\boldsymbol{\xi}$ and $\boldsymbol{\xi}'$ can be used interchangeably since the corresponding control laws gives exactly the same result. In the following, we select the set $\boldsymbol{\xi}$.

C. New Interaction Matrix

The new interaction matrix, which is obtained by stacking the two interaction matrices $\mathbf{L}_{\mathbf{s}_t}$ and $\mathbf{L}_{\boldsymbol{\xi}}$, is a lower block triangular square matrix given by

$$\mathbf{L}_{\mathbf{s_{sp}}} = \begin{bmatrix} \mathbf{L}_{\upsilon} & \mathbf{0} \\ -\mathbf{L}_{\omega}(\theta_{\xi}, \mathbf{u}_{\xi})\mathbf{L}_{\omega,\upsilon} & \mathbf{L}_{\omega}(\theta_{\xi}, \mathbf{u}_{\xi}) \end{bmatrix} \tag{18}$$

where matrix $\mathbf{L}_{\upsilon}$ is given by (5), matrix $\mathbf{L}_{\omega,\upsilon}$ is given by (11), and matrix $\mathbf{L}_{\omega}(\theta_{\xi}, \mathbf{u}_{\xi})$ is given in (15).

Let us mention that if $\mathbf{p_{is}}$ and $\mathbf{p_{js}}$ are the projections of two points lying on the surface of a spherical object, then it is possible to obtain an upper block triangular interaction matrix for this particular target using the $\theta\mathbf{u}$ representation of the relative orientation ${}^{c*}\mathbf{R}_{c}$ computed from (7) [19]. Here, the partial decoupling is obtained from the distances between three image points, which we consider to represent a more general object.

To sum up, the new set $\mathbf{s_{sp}} = (\mathbf{s}_t, \boldsymbol{\xi})$, built using only three points, is such that the decoupling is obtained by the set $\mathbf{s}_t = (d_{01}, d_{02}, d_{12})$; camera rotation motions are controlled using the angle-axis parameterization $\boldsymbol{\xi}$ of a rotation matrix measured from the image of two points. The interaction matrix related to $\boldsymbol{\xi}$, which is proposed here, is valid for a general configuration between two points.

III. Control Analysis

Due to the lower block triangular form of $\mathbf{L}_{\mathbf{s_{sp}}}$, the determination of its singularity domain is shown, in this section, to be easier than in the case of the perspective projection of three points. We also analyze the stability of the control law.

We use the classical control law given by

$$\mathbf{v}_c = -\lambda\widehat{\mathbf{L}_{\mathbf{s_{sp}}}}^{-1}(\mathbf{s_{sp}} - \mathbf{s}^*_{\mathbf{sp}}) \tag{19}$$

where $\mathbf{v}_c = (\boldsymbol{\upsilon}, \boldsymbol{\omega})$ is the camera velocity sent to the low-level robot controller, λ is a positive gain, and

$$\widehat{\mathbf{L}_{\mathbf{s_{sp}}}}^{-1} = \begin{bmatrix} \widehat{\mathbf{L}}_{\upsilon}^{-1} & \mathbf{0} \\ -\widehat{\mathbf{L}}_{\omega,\upsilon}\widehat{\mathbf{L}}_{\upsilon}^{-1} & \widehat{\mathbf{L}}_{\omega}^{-1}(\widehat{\theta}_{\xi}, \widehat{\mathbf{u}}_{\xi}) \end{bmatrix}$$

is the inverse of an approximation of the interaction matrix related to $\mathbf{s_{sp}}$. This approximation takes into account calibration errors, image-processing errors, and modeling errors w.r.t. ranges of points. We first analyze the control law in the ideal case, where the interaction matrix is perfectly known, and then in the case where the range values are approximated.

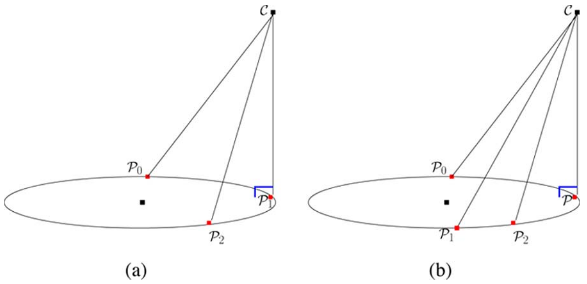


Fig. 3. Cylinder of singularities. (a) Particular case and (b) general case.

A. Ideal Case

Taking the inverse of the interaction matrix (18) and plugging it into (19) leads to the ideal control law

$$\begin{cases} \boldsymbol{\upsilon} = -\lambda\mathbf{L}_{\upsilon}^{-1}(\mathbf{s}_t - \mathbf{s}^*_t) \\ \boldsymbol{\omega} = -\mathbf{L}_{\omega,\upsilon}\boldsymbol{\upsilon} + \lambda\boldsymbol{\xi} \end{cases} \tag{20}$$

since $\mathbf{L}_{\omega}^{-1}(\theta_{\xi}, \mathbf{u}_{\xi})\boldsymbol{\xi} = -\boldsymbol{\xi}$.

The domain of singularity of $\mathbf{L}_{\mathbf{s_{sp}}}$ is given by the following theorem.

Theorem 1: The classical control method (20) is singular if and only if: the three points are aligned, or the camera optical center $\mathcal{C}$ lies on the cylinder of singularities, i.e., the cylinder, which is defined by the circumcircle passing through the three points and the normal to the plane on which the points lie (see Fig. 3).

The proof of Theorem 1 is given in Appendix B. Note that the given proof is easy and simple in comparison with the complex and difficult demonstration given in the case of the perspective coordinates in [23].

Theorem 1 ensures that the new set $\mathbf{s_{sp}}$ does not present more singularities than the perspective coordinates of the three points $\mathbf{s_{pp}} = (p_{0x}, p_{0y}, p_{1x}, p_{1y}, p_{2x}, p_{2y})$, where

$$\mathbf{L}_{\mathbf{s_{pp}}} = [\,\mathbf{L}_t \quad \mathbf{L}_r\,] \tag{21}$$

with

$$\mathbf{L}_t = \begin{bmatrix} -1/P_{0z} & 0 & p_{0x}/P_{0z} \\ 0 & -1/P_{0z} & p_{0y}/P_{0z} \\ -1/P_{1z} & 0 & p_{1x}/P_{1z} \\ 0 & -1/P_{1z} & p_{1y}/P_{1z} \\ -1/P_{2z} & 0 & p_{2x}/P_{2z} \\ 0 & -1/P_{2z} & p_{2y}/P_{2z} \end{bmatrix}$$

and

$$\mathbf{L}_r = \begin{bmatrix} p_{0x}p_{0y} & -(1 + p_{0x}^2) & p_{0y} \\ 1 + p_{0y}^2 & -p_{0x}p_{0y} & -p_{0x} \\ p_{1x}p_{1y} & -(1 + p_{1x}^2) & p_{1y} \\ 1 + p_{1y}^2 & -p_{1x}p_{1y} & -p_{1x} \\ p_{2x}p_{2y} & -(1 + p_{2x}^2) & p_{2y} \\ 1 + p_{2y}^2 & -p_{2x}p_{2y} & -p_{2x} \end{bmatrix}.$$

Indeed, using $\mathbf{s}_{\mathbf{pp}}$, the classical control law (19) (with $\mathbf{s}_{\mathbf{pp}}$ instead of $\mathbf{s}_{\mathbf{sp}}$) has been shown to be singular when the three points are aligned or when the camera optical center belongs to the cylinder of singularities [23].

With the decoupling property of $\mathbf{s}_{\mathbf{sp}}$, it is also easier to determine the kernel of $\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}$ that characterizes the set of camera motions $\mathbf{v}_c \neq \mathbf{0}$, which leaves the image unchanged, i.e., $\dot{\mathbf{s}}_{\mathbf{sp}} = \mathbf{0}$. Indeed, from the expression of the interaction matrix given in (18), we have

$$\ker(\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}) = \{\mathbf{v}_c{=}(\boldsymbol{v}, \boldsymbol{\omega}) \in \mathfrak{se}(\mathbf{3}), \mathbf{L}_{\upsilon}\boldsymbol{v} = \mathbf{0}, \boldsymbol{\omega} = \mathbf{L}_{\omega,\upsilon}\boldsymbol{v}\} \tag{22}$$

which shows that we have to deal only with the 3×3 matrix $\mathbf{L}_{\upsilon}$, contrary to the perspective projection of three points, where we have to deal with the larger and complex 6×6 matrix $\mathbf{L}_{\mathbf{s}_{\mathbf{pp}}}$.

Now, we analyze the control-law stability. We assume, in the following, that the interaction matrix never loses its rank during the servoing, i.e., the camera never crosses the cylinder of singularities, as shown in Fig. 3. Let $\mathbf{e} = \mathbf{s}_{\mathbf{sp}} - \mathbf{s}^*_{\mathbf{sp}}$ be the error to regulate to zero. The closed-loop system equation (using the control law (19) in the ideal case) can be written as $\dot{\mathbf{e}} = -\lambda \mathbf{L}_{\mathbf{s}_{\mathbf{sp}}} \mathbf{L}^{-1}_{\mathbf{s}_{\mathbf{sp}}} \mathbf{e} = -\lambda \mathbf{e}$, which means that the system is locally asymptotically stable (LAS). The system is not globally stable since we assumed the camera never crosses the cylinder of singularities; in addition, since with three points, four different camera poses exist providing the same points position in the image [21], reaching the desired pose is only possible from a local neighborhood.

On one hand, the set $\mathbf{s}_{\mathbf{sp}}$ is perfectly suited to control camera rotations compared with the coordinates of the center of gravity used in [18]. Indeed, for a pure rotation motion, since the value of $\mathbf{s}_t$ is constant, i.e., $\mathbf{s}_t = \mathbf{s}^*_t$, the expression of the control law (20) is given by

$$\boldsymbol{v} = \mathbf{0}, \qquad \boldsymbol{\omega} = \lambda \boldsymbol{\xi} \tag{23}$$

which means that the camera center follows a geodesic path. On the other hand, as for classical perspective coordinates of points as soon as all points do not have the same depth, one disadvantage of the proposed approach is that a pure translation displacement also generates rotation motions. Indeed, from (20), it is clear that, even if $\boldsymbol{\xi} = 0, \boldsymbol{\omega} \neq 0$ as soon as $\mathbf{s}_t \neq \mathbf{s}^*_t$. However, we will see in Section IV-A and B that the results obtained, in practice, for a pure translation motion do not deviate a lot from a straight-line trajectory in the Cartesian space and in the image space.

B. Approximation of Range Values

We now consider the usual case, in practice, where the true value of $\|\mathbf{P}_i\|$ is not available. In that case, the closed-loop system equation using the control law (19) can be written as

$$\dot{\mathbf{e}} = -\lambda \mathbf{L}_{\mathbf{s}_{\mathbf{sp}}} \widehat{\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}}^{-1} \mathbf{e} \tag{24}$$

with

$$\widehat{\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}}^{-1} = \begin{bmatrix} \widehat{\mathbf{L}}_{\upsilon}^{-1} & \mathbf{0} \\ -\widehat{\mathbf{L}}_{\omega,\upsilon}\widehat{\mathbf{L}}_{\upsilon}^{-1} & \mathbf{L}_{\omega}^{-1}(\theta_{\xi}, \mathbf{u}_{\xi}) \end{bmatrix}$$

where $\widehat{\mathbf{L}}_{\upsilon}^{-1}$ and $\widehat{\mathbf{L}}_{\omega,\upsilon}$ depend on $\widehat{\|\mathbf{P}_i\|}, i = 0, 1, 2$. $\widehat{\|\mathbf{P}_i\|}$, which is the estimated value of $\|\mathbf{P}_i\|$, can be expressed as follows:

$$\widehat{\|\mathbf{P}_i\|} = |\widehat{P_{iz}}|\rho_i \tag{25}$$

with $\rho_i = \sqrt{(P_{ix}/P_{iz})^2 + (P_{iy}/P_{iz})^2 + 1}$, where $P_{ix}/P_{iz} = p_{six}/p_{siz}$, and $P_{iy}/P_{iz} = p_{siy}/p_{siz}$ are measured from the image of points. The interaction matrix (18) thus depends on the estimated depth value $\widehat{P_{iz}}$.

In the case, where $|\widehat{P_{iz}}| = \alpha |P_{iz}|, i = 0, 1, 2$, it is easy to show that the system is LAS for $\alpha > 0$ and the singularity domain of $\widehat{\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}}(\widehat{P_{iz}})$ is exactly the singularity domain of $\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}(P_{iz})$ since $\widehat{\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}}(\widehat{P_{iz}}) = (1/\alpha)\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}(P_{iz})$. This control robustness analysis is validated by the last simulation result presented in Section IV-A.

In the general case, the singularity and the stability analysis of the control using $\widehat{\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}}(\widehat{P_{iz}})$ are complex to establish and are left for future developments. Using a very coarse approaximation of the depth value $\widehat{P_{iz}} = P^*_{iz}$, the control law built with either $\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}(P^*_{iz})$ or $\mathbf{L}_{\mathbf{s}^*_{\mathbf{sp}}}(P^*_{iz})$ is well-known to be LAS [1]. In practice, the validation of the use of $\mathbf{L}_{\mathbf{s}_{\mathbf{sp}}}(P^*_{iz})$ and $\mathbf{L}_{\mathbf{s}^*_{\mathbf{sp}}}(P^*_{iz})$ is shown in Section IV-B with perspective and fish-eye cameras, respectively.

Finally, in the case of pure rotations, it is clear from (19) that the control law is perfectly adequate, even in presence of errors on points depths since the depths of points are not involved. Indeed, we still have in that case $\boldsymbol{v} = \mathbf{0}$, and $\boldsymbol{\omega} = \lambda\boldsymbol{\xi}$, as in (23).

To sum up, as a benefit compared with the use of classical perspective image of points and first-order spherical moments [18], the new set has been theoretically shown to be optimal for camera rotation motions, even in the case of depths errors. In ideal conditions and, to some extent, in the case of depth errors, the singularity domain of a classical control using the new set of features has been theoretically characterized and shown not to be larger than the singularity domain of the classical perspective coordinates of points given in [23]; the partial decoupling of the control thus makes the new set able to provide adequate camera motion in comparison with the classical perspective coordinates of points, as confirmed by results given in Section IV-A. Even though the stability of the control law is complex to analyze in the general case of depth errors, the control is LAS when using the desired depth values.

IV. Results

This section illustrates the proposed theoretical results through simulations and real experiments. In simulation results, we assume that the point depths values are perfectly known. In first real experiments, using the current interaction matrix with a perspective camera, the point depths are coarsely approximated by their desired values. Finally, the LAS property is also validated using the desired interaction matrix with a fish-eye camera.

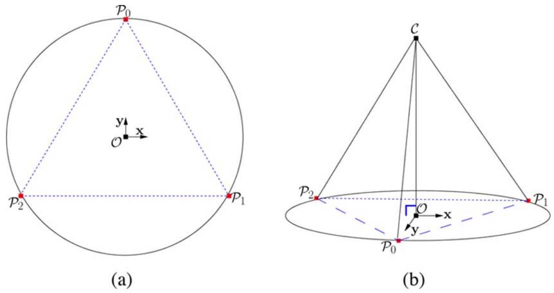


Fig. 4. Configuration of three points in the 3-D space. (a) Equilateral triangle and circumcircle and (b) desired pose of the camera.

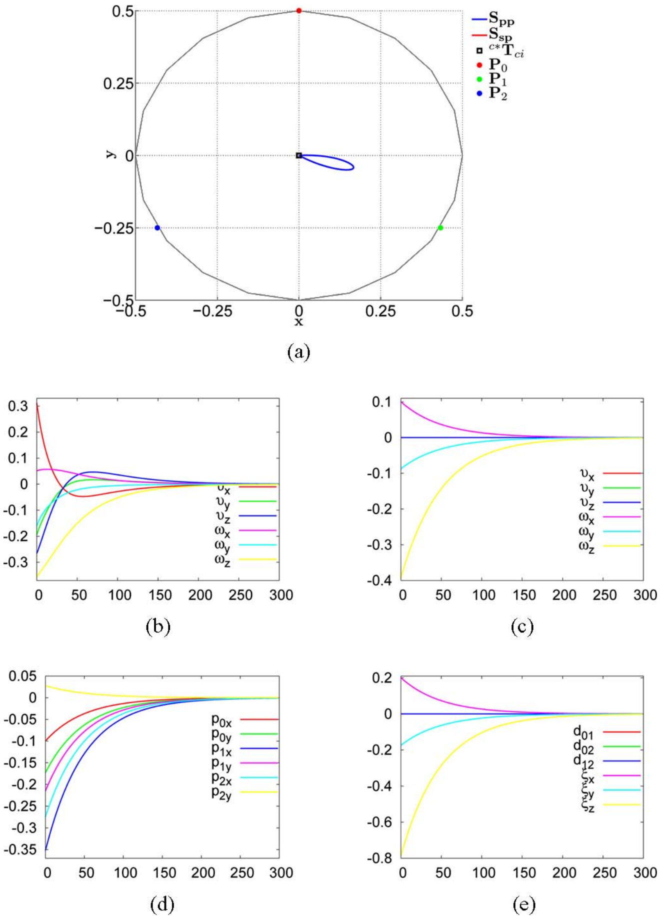


Fig. 5. Decoupled versus coupled control. (a) Camera Cartesian trajectories, (b) and (c) computed camera velocities (in meters per second and radians per second) using $\mathbf{s_{pp}}$ and $\mathbf{s_{sp}}$, and (d) and (e) errors on $\mathbf{s_{pp}}$ spp and $\mathbf{s_{sp}}$.

A. Simulation Results

In this section, we compare the new set $\mathbf{s_{sp}}$ with the classical perspective coordinates of three points $\mathbf{s_{pp}}$ using the simulator provided in ViSP [28].

Note that since at least four points are required to determine either the set of features proposed in [18] or the relative camera pose computed from a homography transformation in [13], it is not possible to compare these methods with the new set of six features, which uses only three points. This comparison will be done in future works in which we plan to extend the present work to the case of more than three points.

We first present the case of camera rotation motions only; then, we consider only translation motions; finally, we present the case where both camera rotation and translation motions are involved.

In the object frame $\mathcal{F}_o = (\mathcal{O}, \mathbf{x}, \mathbf{y}, \mathbf{z})$, the coordinates of the three points are given by $\mathbf{P}_0 = r(0, 1, 0.30)$, $\mathbf{P}_1 = r(\sqrt{3}/2, -1/2, 0.30)$, and $\mathbf{P}_2 = r(-\sqrt{3}/2, -1/2, 0.30)$, with $r = 0.5$ m. The configuration of the three points describes an equilateral triangle, as shown in Fig. 4(a), and the circumcircle (of center $\mathcal{O}$ and radius r) of the equilateral triangle is the circular cross section of the cylinder of singularities. The pose of the object frame $\mathcal{F}_o$ w.r.t. the desired pose of the camera frame $\mathcal{F}_{c*}$ is set to the values $^{c*}\mathbf{t}_o = (0, 0, 2.35)$ m and $\theta\mathbf{u}(^{c*}\mathbf{R}_o) = (0, 0, 0)$ rad, which means that the desired pose of the camera is inside the cylinder of singularities, as shown in Fig. 4(b).

Now, we present three experiments, where we consider only camera rotation motions. In the first experiment, we highlight both the partial decoupling of the control using the new set $\mathbf{s_{sp}}$ and the coupling of the control using the classical set $\mathbf{s_{pp}}$. The orientation of the initial camera frame w.r.t. the desired camera frame has been set to $\theta\mathbf{u}(^{c*}\mathbf{R}_c) = (-0.20, 0.17, 0.79)$ rad. As expected, because of the partial decoupling, the new set $\mathbf{s_{sp}}$ causes no translation displacement of the camera, while the classical set $\mathbf{s_{pp}}$ does [see Fig. 5(a)]. Indeed, the coupling in the control law using the classical set $\mathbf{s_{pp}}$ [see (21)] generates undesired translation velocities with oscillations [compare Fig. 5(b) and (c)]. Using the classical set, all the features vary, which is not the case when using the new set where only the subset $\boldsymbol{\xi}$ varies since the subset (d_{01}, d_{02}, d_{12}) is invariant to rotations [compare Fig. 5(d) and (e)]. As expected also, the control using the new set $\mathbf{s_{sp}}$ is ideal since the camera rotation velocities are linearly linked to the set of features $\boldsymbol{\xi}$, and both decrease exponentially [compare Fig. 5(e) and (c)].

In the second experiment, we show the superiority of the new set $\mathbf{s_{sp}}$ over the classical set $\mathbf{s_{pp}}$ in the case of large rotation motions of the camera. The orientation of the initial camera frame w.r.t. the desired camera frame has been set to $\theta\mathbf{u}(^{c*}\mathbf{R}_c) = (0.21, -0.31, 0.47)$ rad. In this case, the control with the classical set $\mathbf{s_{pp}}$ crosses the cylinder of singularities and converges toward another global minimum [see Fig. 6(a)]. Using the classical set, visual servoing fails because of the translation motions induced by the coupling in $\mathbf{L}_{\mathbf{s_{pp}}}$, whereas, as expected, the control using $\mathbf{s_{sp}}$ converges ideally and exponentially (as before), as shown in Fig. 6(b) and (c).

In the third experiment, we validate the robustness of the control in the case of modeling errors, i.e., errors on points depths. We have introduced the following error of the depth estimation of points: $\widehat{P_{0z}} = 0.5P_{0z}$, $\widehat{P_{1z}} = 0.3P_{1z}$, and $\widehat{P_{2z}} = 0.75P_{2z}$. The relative orientation of the initial camera frame w.r.t. the desired camera frame is set to the same value, as in the second experiment described above. As expected, the robot displays exactly the same ideal behavior, as shown in Fig. 6(b) and (c), where only rotation motions are generated. Indeed, even in the

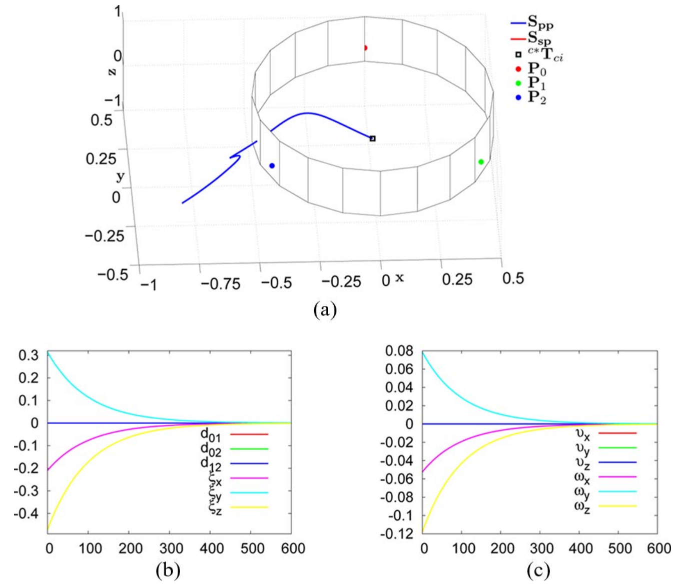


Fig. 6. Superiority for rotationmotions. (a) Cylinder of singularities is crossed by the control using the classical set $\mathbf{s_{pp}}$ and (b) and (c) errors and computed camera velocities (in meters per second and radians per second) using $\mathbf{s_{sp}}$.

case of modeling errors, the control scheme generates rotation motions only, as given by the expression (23).

Now, we validate the new set for a translation motion and compare it with the classical set. The relative pose of the initial camera frame w.r.t. the desired camera frame is set to the value $^{c*}\mathbf{t}_c = (0.21, 0.31, -0.5)$ m. Using the new set, even if there is a little oscillation on $\boldsymbol{v}_x$ (at the beginning of the servoing), which does not appear with the control using the classical set [compare Fig. 7(b) and (c)], the robot Cartesian trajectory is satisfactory [see Fig. 7(a)]. As expected, in both cases, all visual features vary and the errors decrease exponentially, as shown in Fig. 7(d) and (e).

In the next three experiments, we consider complex motions, i.e., motions made up of both rotation and translation displacements. In the first experiment, we validate the new set in the case of a motion, where the relative pose of the initial camera frame w.r.t. the desired camera frame is set to the following value: $^{c*}\mathbf{t}_c = (0.29, 0.16, -0.52)$ m and $\theta\mathbf{u}(^{c*}\mathbf{R}_c) = (0.21, -0.21, 0.30)$ rad. In this case, we have a better Cartesian trajectory (straight line) using the new set [see Fig. 8(a)]. In addition, camera velocities present no oscillation contrary to the velocities using the classical set [compare Fig. 8(b) and (c)].

In the second experiment, in order to validate the larger convergence domain of the control using the new set, we consider a relatively large displacement where the camera initial pose is very close to the boundary of the cylinder of singularities. The relative initial pose of the camera frame w.r.t. the desired camera frame is set to the following value: $^{c*}\mathbf{t}_c = (-0.29, -0.37, -0.48)$ m and $\theta\mathbf{u}(^{c*}\mathbf{R}_c) = (-0.26, 0.17, -0.52)$ rad. This particular position, which is close to a singularity of the control, induces a large condition number value (which is almost 70) of the block matrix $\mathbf{L}_v$, as shown at the beginning of the servo in Fig. 9(a). The coupled control induced by the classical set causes the control to cross the cylinder of singularities and to converge toward another global

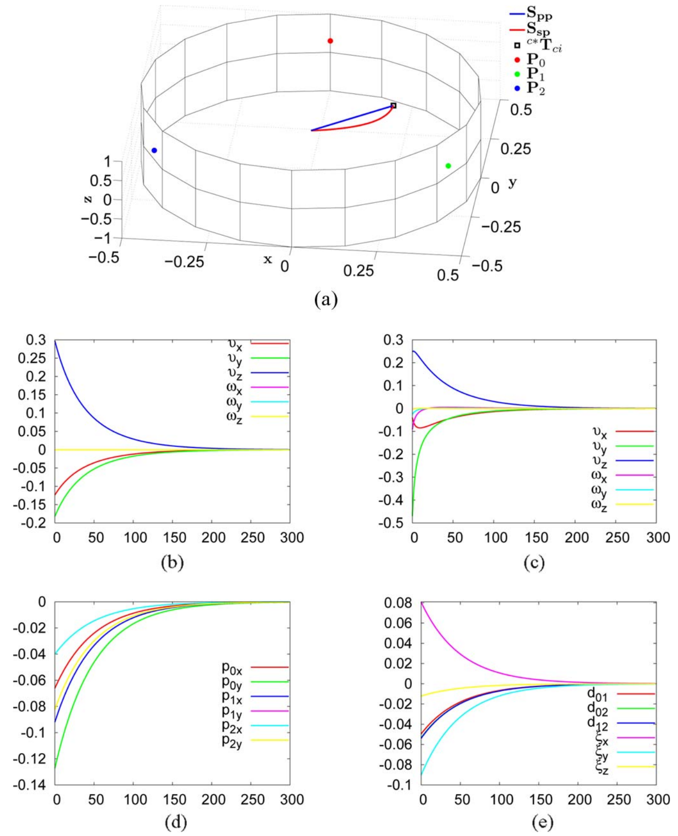


Fig. 7. Comparison in the case of a translation motion. (a) Camera Cartesian trajectories, (b) and (c) computed camera velocities (in meters per second and radians per second) using $\mathbf{s_{pp}}$ and $\mathbf{s_{sp}}$, and (d) and (e) errors on $\mathbf{s_{pp}}$ and $\mathbf{s_{sp}}$.

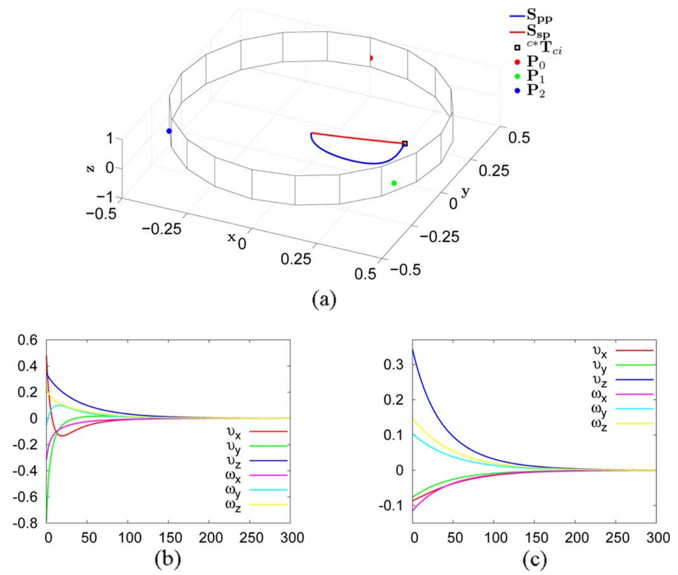


Fig. 8. Comparison in the case of a complex motion. (a) Camera Cartesian trajectories and (b) and (c) computed camera velocities (in meters per second and radians per second) using $\mathbf{s_{pp}}$ and $\mathbf{s_{sp}}$.

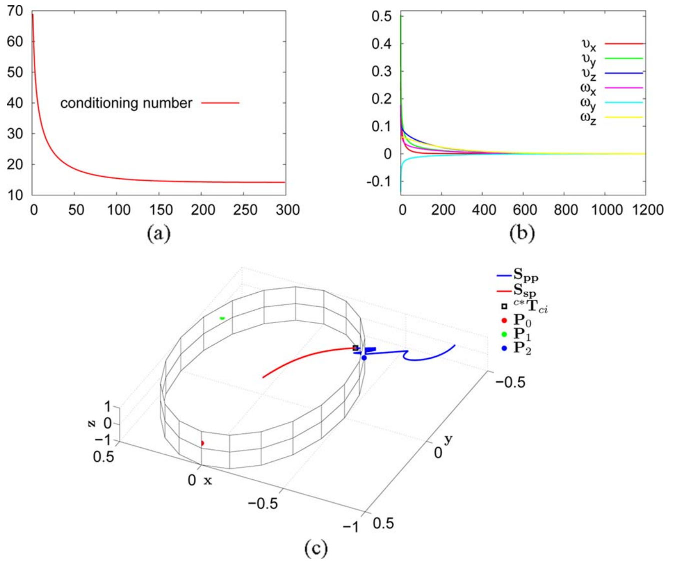


Fig. 9. Larger convergence domain for the new set, where the cylinder of singularities is crossed by the classical set $\mathbf{s_{pp}}$. (a) Conditioning number of the block matrix $\mathbf{L}_\upsilon$, (b) computed camera velocities (in meters per second and radians per second), and (c) camera Cartesian trajectories.

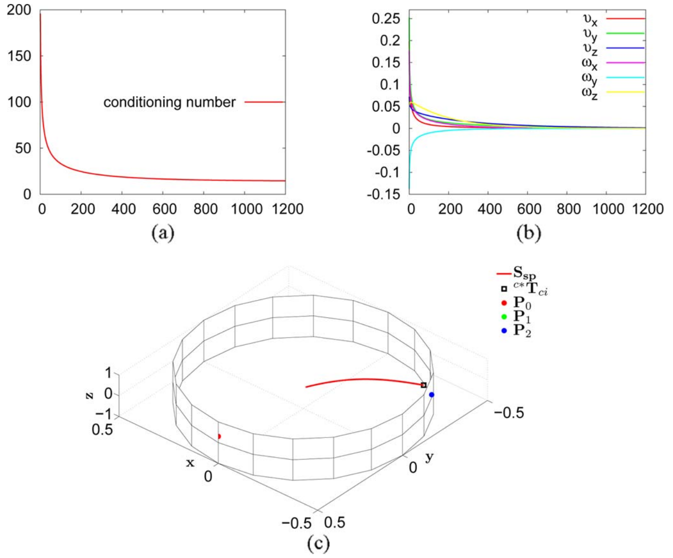


Fig. 10. Robustness w.r.t. modeling errors using the new set $\mathbf{s_{sp}}$. (a) Conditioning number of the block matrix $\mathbf{L}_\upsilon$, (b) computed camera velocities (in meters per second and radians per second), and (c) camera Cartesian trajectories.

minimum [see Fig. 9(c)], while the new control shows satisfactory Cartesian trajectory and converges toward the desired pose [for the camera velocities, see Fig. 9(b)].

In the third experiment, we validate the robustness of the new control to modeling errors. We have set the following error on the depth estimation of points: $\widehat{P_{iz}} = 0.5P_{iz}, i = 0, 1, 2$. The camera is initially positioned exactly the same as in the last experiment, i.e., near to the boundary of the cylinder of singularities. The results shown in Fig. 10(b) illustrate the control law convergence, despite a very high conditioning number (which is almost 200), due to modeling errors, of the block matrix $\mathbf{L}_\upsilon$ at the beginning of the servo, as shown in Fig. 10(a); in addition, the robot Cartesian trajectory is satisfactory, as shown in Fig. 10(c).

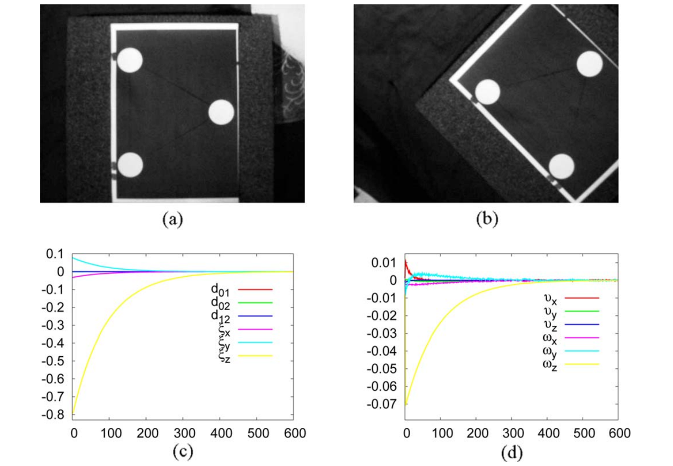


Fig. 11. Rotation displacement. (a) Desired image, (b) initial image, (c) errors, and (d) computed camera velocities (in meters per second and radians per second).

To conclude, simulation results have shown that the new set is ideal for camera rotation motions; more importantly, using the new set, the convergence domain has been shown to be larger than in the case of the classical perspective coordinates.

B. *Experimental Results*

Now, we present five experiments that validate the use of our approach in practice. We use a 6-DOF robot arm equipped either with a perspective camera or with a fish-eye camera. The first four experiments use a perspective camera: the central point is located at $(u_0, v_0) = (320, 240)$ pixels and the couple of focal length values is given by $(f_u, f_v) = (600, 600)$ pixels/m. Experiments with that perspective camera have been carried out using $\widehat{\mathbf{L}_{\mathbf{s_{sp}}}}(\widehat{P_{iz}} = P_{iz}^*)$ in the control law (19). In addition, the desired pose of the perspective camera is the same as for simulation results [see Fig. 4(b)], i.e., the desired camera pose is such that the image plane is parallel to the plane defined by the three points. In this case, using Thalès theorem with triangles $(\mathcal{C}, \mathcal{P}_0, \mathcal{P}_1)$, $(\mathcal{C}, \mathbf{p_{0s}}, \mathbf{p_{1s}})$, and (25), the estimated value of the common desired depth P_z^* of points is given by

$$P_z^* = \frac{\|\mathbf{P_1^*} - \mathbf{P_0^*}\|}{\rho_0^* \|\mathbf{p_{1s}^*} - \mathbf{p_{0s}^*}\|}.$$

In the first experiment, we consider a rotation between the current and the desired camera frames: $^{c*}\mathbf{t}_c = (0.0, 0.0, 0.0)$ m and $\theta\mathbf{u}(^{c*}\mathbf{R}_c) = (5°, -7°, 45°)$. The desired and initial images are shown in Fig. 11(a) and (b), respectively. Fig. 11(c) shows the errors, and Fig. 11(d) shows the robot velocities, where we can see unexpectedly very small translation velocities. These translation velocities are caused by hand–eye calibration errors. Despite these unexpected small translation velocities, the convergence of the control proves the robustness of the new scheme w.r.t. calibration errors.

In the second experiment, we consider only a translation displacement: $^{c*}\mathbf{t}_c = (-0.1, 0.05, -0.2)$ m and $\theta\mathbf{u}(^{c*}\mathbf{R}_c) =$

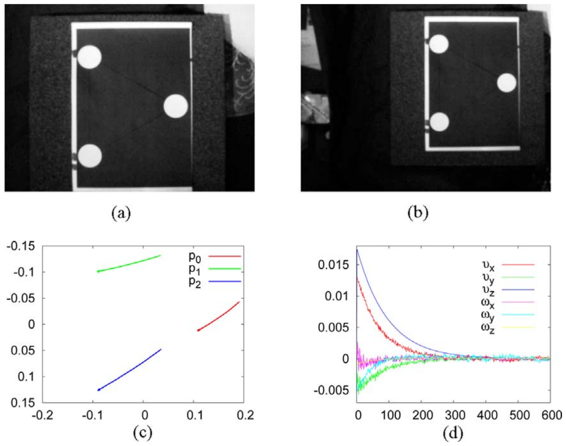


Fig. 12. Translation displacement. (a) Desired image, (b) initial image, (c) image trajectories, and (d) computed camera velocities (in meters per second and radians per second).

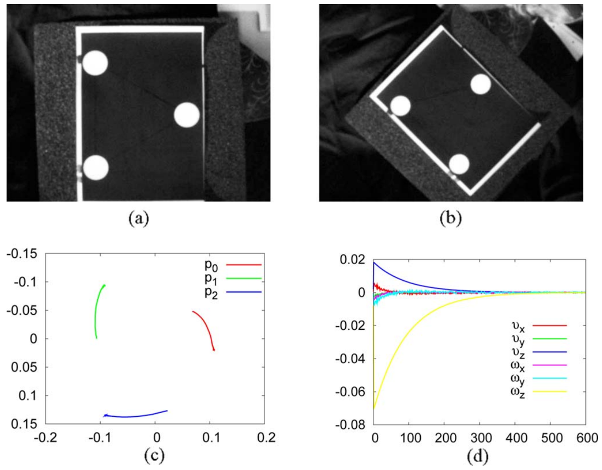


Fig. 13. 45° rotation around and 20-cm translation along the optical axis. (a) Desired image, (b) initial image, (c) image trajectories, and (d) computed camera velocities (in meters per second and radians per second).

$(0.0°, 0.0°, 0.0°)$. The desired and initial images are shown in Fig. 12(a) and (b), respectively. The obtained results show image trajectories in Fig. 12(c) and the convergence of the control despite little noise on velocities, as shown in Fig. 12(d). Image trajectories are almost straight lines, which are satisfactory.

The third experiment illustrated in Fig. 13 shows, as expected, that our new control scheme performs adequately in the case of a rotation around the optical axis and a translation along the optical axis.

In the fourth experiment, we consider a rotation and a translation displacement between the current and the desired camera frames: ${}^{c*}\mathbf{t}_c = (0.0, -0.05, -0.2)$ m and $\theta\mathbf{u}({}^{c*}\mathbf{R}_c) = (5.0°, -7.0°, 15.0°)$. The desired and initial images are shown in Fig. 14(a) and (b), respectively. Fig. 14(d) shows the velocities and Fig. 14(c) shows the Cartesian trajectory, where we can see, once more, that the control converges satisfactorily.

Finally, we validate once more the control LAS property using an omnidirectional vision system, that is a fish-eye camera [see Fig. 15(a) and (b)]; the mirror-like parameter is $d_f = 1.71$, the central point is located at $(u_0, v_0) = (315.61, 243.05)$ pixels, and the couple of focal length values is given by $(f_u, f_v) = (722.91, 721.65)$ pixels/m. We use the desired value $\mathbf{L}_{\mathbf{s}^*_{\mathrm{sp}}}(P^*_{iz})$ of the interaction matrix in the control law (19). Note that although the target is composed of four points, only three points are used for visual servoing. Fig. 15(c) and (d) shows the result, where we can see that the control converges once more with a satisfactory behavior.

To conclude, experimental results have demonstrated successful performances of the proposed control scheme, even in the case of calibration and modeling errors on points depths.

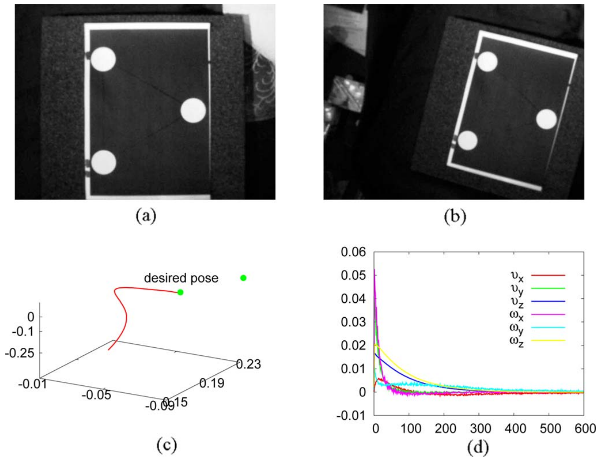


Fig. 14. Rotation and translation displacement. (a) Desired image, (b) initial image, (c) Cartesian trajectory, and (d) computed camera velocities (in meters per second and radians per second).

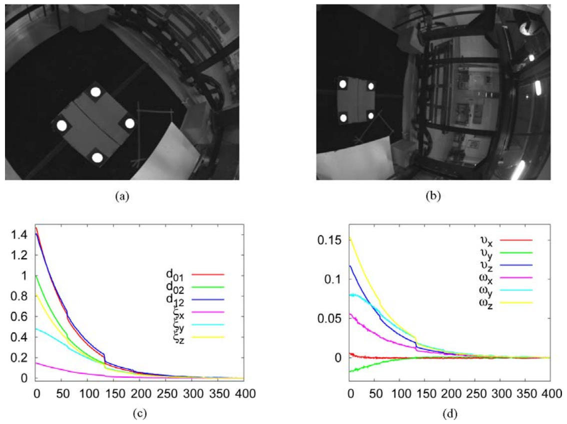


Fig. 15. Rotation and translation displacement. (a) Desired image, (b) initial image, (c) errors, and (d) computed camera velocities (in meters per second and radians per second).

V. Conclusion and Future Works

In this paper, we have proposed a new set of six features for visual servoing from three points using a spherical-projection model. This new set includes distances between spherical projections of points, which allow to obtain a partially decoupled control scheme. In comparison with the classical perspective coordinates of three points, we have demonstrated that the new set does not present more singularities. Indeed, the singularity domain of the new set has been theoretically characterized. The singularities appear when either the three points are aligned or the camera optical center lies on the well-known cylinder of singularities. Using the new set, it is important to mention the simplicity of the determination of the cylinder of singularities in comparison with the complex demonstration given in the case of the perspective coordinates. The new set has been formally shown to be optimal for camera rotation motions compared with the classical perspective coordinates of points and the modern set using spherical moments. These theoretical results have been successfully validated through simulations and real experiments. Simulation results have shown that the convergence domain of a classical control method using the new set is larger than with the classical set. Experimental results on a 6-DOF robotic system have validated the use of our approach in the case of calibration errors and modeling errors on points depths. As future works, it will be interesting to extend the distance-based control scheme to the case of more than three points and to determine either a set of six visual features or a redundant set of features that provides decoupling properties. In the first case, the singularity domain of the control is the main problem, whereas in the second case, it will be necessary to find a set free of local minima.

Appendix A

Computation of the Tangent Vector $\boldsymbol{\zeta}$

Here, the detailed developments of the computation of the tangent vector $\boldsymbol{\zeta}$ to ${}^{c}\mathbf{R}_{p_{is}}$ in $\mathbf{SO(3)}$ are presented using the framework proposed in [22].

From (8), we immediately obtain

$$\boldsymbol{\zeta} = \left[{}^{c}\dot{\mathbf{R}}_{p_{is}}{}^{p_{is}}\mathbf{R}_{c}\right]^{\times} \tag{26}$$

where $[\mathbf{M}]^{\times}$ is the vector associated with the antisymmetric matrix $\mathbf{M}$. Using the property

$$\mathbf{S}\mathbf{w} = \left[\mathbf{S}[\mathbf{w}]_{\times}\mathbf{S}^{\top}\right]^{\times} \quad \forall\mathbf{S}\in\mathbf{SO(3)} \quad \text{and} \quad \forall\mathbf{w}\in\mathbb{R}^{3} \tag{27}$$

with settings $\mathbf{S} = {}^{p_{is}}\mathbf{R}_{c}$, and $\mathbf{w} = [{}^{c}\dot{\mathbf{R}}_{p_{is}}{}^{p_{is}}\mathbf{R}_{c}]^{\times}$, from (26), we easily obtain

$$\boldsymbol{\zeta} = {}^{c}\mathbf{R}_{p_{is}}[{}^{p_{is}}\mathbf{R}_{c}{}^{c}\dot{\mathbf{R}}_{p_{is}}]^{\times}. \tag{28}$$

We have

$${}^{p_{is}}\mathbf{R}_{c}{}^{c}\dot{\mathbf{R}}_{p_{is}} = \begin{bmatrix} 0 & \mathbf{v_1}^{\top}\dot{\mathbf{v}}_{\mathbf{2}} & \mathbf{v_1}^{\top}\dot{\mathbf{v}}_{\mathbf{3}} \\ \mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{1}} & 0 & \mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{3}} \\ \mathbf{v_3}^{\top}\dot{\mathbf{v}}_{\mathbf{1}} & \mathbf{v_3}^{\top}\dot{\mathbf{v}}_{\mathbf{2}} & 0 \end{bmatrix}$$

which leads to

$$[{}^{p_{is}}\mathbf{R}_{c}{}^{c}\dot{\mathbf{R}}_{p_{is}}]^{\times} = \begin{bmatrix}-\mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{3}} & -\mathbf{v_3}^{\top}\dot{\mathbf{v}}_{\mathbf{1}} & \mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{1}}\end{bmatrix}^{\top}. \tag{29}$$

Note that the component of vector $\left[{}^{p_{is}}\mathbf{R}_{c}{}^{c}\dot{\mathbf{R}}_{p_{is}}\right]^{\times}$ has been chosen in order to simplify the computations. By plugging (29) into (28), we obtain

$$\boldsymbol{\zeta} = \begin{bmatrix}\mathbf{v_1} & \mathbf{v_2} & \mathbf{v_3}\end{bmatrix}\begin{bmatrix}-\mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{3}} \\ -\mathbf{v_3}^{\top}\dot{\mathbf{v}}_{\mathbf{1}} \\ \mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{1}}\end{bmatrix}. \tag{30}$$

Since vector $\mathbf{v_1}$ is the spherical projection of the point $\mathcal{P}_i$, from (3), we easily deduce the time variation of $\mathbf{v_1}$ as

$$\dot{\mathbf{v}}_{\mathbf{1}} == -\frac{1}{\|\mathbf{P}_i\|}\left(\mathbf{I}_3 - \mathbf{v_1}\mathbf{v_1}^{\top}\right)\boldsymbol{v} + \mathbf{v_1}\times\omega. \tag{31}$$

From (31), we obtain after some developments

$$\mathbf{v_3}^{\top}\dot{\mathbf{v}}_{\mathbf{1}} = -\frac{1}{\|\mathbf{P}_i\|}\mathbf{v_3}^{\top}\boldsymbol{v} + \mathbf{v_2}^{\top}\boldsymbol{\omega} \tag{32}$$

and

$$\mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{1}} = -\frac{1}{\|\mathbf{P}_i\|}\mathbf{v_2}^{\top}\boldsymbol{v} - \mathbf{v_3}^{\top}\boldsymbol{\omega}. \tag{33}$$

Now, we present the detailed developments of the expression $\mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{3}}$. It worth mentioning that the computation of $\mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{3}}$ is a contribution of this paper since this computation is valid for a general configuration of points [see Fig. 2(b)], including for the specific configuration between points [see Fig. 2(a)] in [22]. From the expression of $\mathbf{v_3}$ given in (7), we have

$$\dot{\mathbf{v}}_{\mathbf{3}} = \dot{\mathbf{v}}_{\mathbf{1}}\times\mathbf{v_2} + \mathbf{v_1}\times\dot{\mathbf{v}}_{\mathbf{2}}. \tag{34}$$

From (34), we immediately have

$$\mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{3}} = \mathbf{v_2}^{\top}(\mathbf{v_1}\times\dot{\mathbf{v}}_{\mathbf{2}}), \tag{35}$$

since $\mathbf{v_2}^{\top}(\dot{\mathbf{v}}_{\mathbf{1}}\times\mathbf{v_2}) = \dot{\mathbf{v}}_{\mathbf{1}}^{\top}(\mathbf{v_2}\times\mathbf{v_2}) = 0$. From the expression of $\mathbf{v_2}$ given in (7), we obtain after some developments

$$\mathbf{v_1}\times\dot{\mathbf{v}}_{\mathbf{2}} = \frac{1}{\|\mathbf{u}_{\mathbf{p_{i_s}p_{j_s}}}\|}(\mathbf{v_1}\times\dot{\mathbf{u}}_{\mathbf{p_{i_s}p_{j_s}}}) - \frac{(\dot{\|\mathbf{u}_{\mathbf{p_{i_s}p_{j_s}}}\|})}{\|\mathbf{u}_{\mathbf{p_{i_s}p_{j_s}}}\|}\mathbf{v_3} \tag{36}$$

where $\mathbf{u}_{\mathbf{p_{i_s}p_{j_s}}} = \Gamma_{\mathbf{p_{i_s}}}(\mathbf{p_{j_s}} - \mathbf{p_{i_s}})$. By substituting (36) into (35), we obtain

$$\mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{3}} = -\frac{1}{\|\mathbf{u}_{\mathbf{p_{i_s}p_{j_s}}}\|}\mathbf{v_3}^{\top}\dot{\mathbf{u}}_{\mathbf{p_{i_s}p_{j_s}}}. \tag{37}$$

The detailed expression of $\mathbf{u}_{\mathbf{p_{i_s}p_{j_s}}}$ given by

$$\mathbf{u}_{\mathbf{p_{i_s}p_{j_s}}} = \Gamma_{\mathbf{p_{i_s}}}(\mathbf{p_{j_s}} - \mathbf{p_{i_s}}) = \mathbf{p_{j_s}} - \mathbf{p_{i_s}}(\mathbf{p_{i_s}}^{\top}\mathbf{p_{j_s}}) \tag{38}$$

which leads to

$$\dot{\mathbf{u}}_{\mathbf{p_{i_s}p_{j_s}}} = \dot{\mathbf{p}}_{\mathbf{j_s}} - \dot{\mathbf{p}}_{\mathbf{i_s}}(\mathbf{p_{i_s}}^{\top}\mathbf{p_{j_s}}) - \mathbf{p_{i_s}}\left(\mathbf{p_{j_s}}^{\top}\dot{\mathbf{p}}_{\mathbf{i_s}} + \mathbf{p_{i_s}}^{\top}\dot{\mathbf{p}}_{\mathbf{j_s}}\right). \tag{39}$$

From (39), we obtain after some developments

$$\dot{\mathbf{u}}_{\mathbf{p_{i_s}p_{j_s}}} = \mathbf{M}_{\mathbf{p_{i_s}p_{j_s}}}\boldsymbol{v} + \mathbf{u}_{\mathbf{p_{i_s}p_{j_s}}}\times\boldsymbol{\omega} \tag{40}$$

with

$$\begin{aligned}\mathbf{M}_{\mathbf{p_{i_s}p_{j_s}}} &= -\frac{1}{\|\mathbf{P}_j\|}\Gamma_{\mathbf{p_{i_s}}}\Gamma_{\mathbf{p_{j_s}}} \\ &\quad + \frac{1}{\|\mathbf{P}_i\|}\left((\mathbf{p_{i_s}}^{\top}\mathbf{p_{j_s}})\mathbf{I}_3 + \mathbf{p_{i_s}}\mathbf{p_{j_s}}^{\top}\right)\Gamma_{\mathbf{p_{i_s}}}.\end{aligned} \tag{41}$$

By substituting (40) into (37), we obtain

$$\mathbf{v_2}^{\top}\dot{\mathbf{v}}_{\mathbf{3}} = -\frac{1}{\|\Gamma_{\mathbf{p_{i_s}}}(\mathbf{p_{j_s}} - \mathbf{p_{i_s}})\|}\mathbf{v_3}^{\top}\mathbf{M}_{\mathbf{p_{i_s}p_{j_s}}}\boldsymbol{v} + \mathbf{v_1}^{\top}\boldsymbol{\omega}. \tag{42}$$

By expanding the expression of $\mathbf{v_3}$ given in (7), we obtain $\mathbf{v_3} = \frac{1}{\|\Gamma_{\mathbf{p_{i_s}}}(\mathbf{p_{j_s}} - \mathbf{p_{i_s}})\|}(\mathbf{p_{i_s}}\times\mathbf{p_{j_s}})$ from which it is straightforward to get

$$\mathbf{v_3}^\top \mathbf{M}_{\mathbf{p_{i_s}}\mathbf{p_{j_s}}} = \frac{1}{\|\mathbf{P}_i\|} \frac{\left(\|\mathbf{P}_j\|(\mathbf{p_{i_s}}^\top \mathbf{p_{j_s}}) - \|\mathbf{P}_i\|\right)}{\|\mathbf{P}_j\|} \mathbf{v_3}^\top \tag{43}$$

i.e., $\mathbf{v_3}^\top$ is a left eigenvector of matrix $\mathbf{M}_{\mathbf{p_{i_s}}\mathbf{p_{j_s}}}$. Using (43), expression given in (42) simplifies to

$$\mathbf{v_2}^\top \dot{\mathbf{v}}_\mathbf{3} = -\frac{\delta_p}{\|\mathbf{P}_i\|} \mathbf{v_3}^\top \boldsymbol{v} + \mathbf{v_1}^\top \boldsymbol{\omega} \tag{44}$$

with $\delta_p = \frac{(\mathbf{p_{i_s}}^\top \mathbf{p_{j_s}})\|\mathbf{P}_j\| - \|\mathbf{P}_i\|}{\|\mathbf{P}_j\| \|\Gamma_{\mathbf{p_{i_s}}}(\mathbf{p_{j_s}} - \mathbf{p_{i_s}})\|}$. Finally, by substituting (32), (33), and (44) into (30), we obtain

$$\boldsymbol{\zeta} = \mathbf{L}_{\upsilon,\omega} \boldsymbol{v} - \boldsymbol{\omega} \tag{45}$$

with

$$\mathbf{L}_{\omega,\upsilon} = \frac{1}{\|\mathbf{P}_i\|}(\delta_p \mathbf{v_1}\mathbf{v_3}^\top + \mathbf{v_2}\mathbf{v_3}^\top - \mathbf{v_3}\mathbf{v_2}^\top).$$

APPENDIX B

PROOF OF THEOREM 1

Here, we give a proof of Theorem 1.

Proof: The key element to the determination of the singularities of the classical control given in (20) is the factorization of the determinant of the interaction matrix given in (18).

From (18), since $\mathbf{L}_{\mathbf{s_{sp}}}$ is a square triangular matrix, it is immediate to show that

$$\left|\mathbf{L}_{\mathbf{s_{sp}}}\right| = |\mathbf{L}_\upsilon| \, |\mathbf{L}_\omega(\theta_\xi, \mathbf{u}_\xi)|. \tag{46}$$

Since in the case where ${}^c\mathbf{R}_{p_{is}}$ is defined, i.e., when $\mathcal{C}$, $\mathcal{P}_1$, and $\mathcal{P}_2$ are not aligned, $|\mathbf{L}_\omega(\theta_\xi, \mathbf{u}_\xi)| = -1/\text{sinc}^2(\theta_\xi/2)$, $\mathbf{L}_\omega(\theta_\xi, \mathbf{u}_\xi)$ is nonsingular in the task space [12]. Therefore, we focus on the factorization of $|\mathbf{L}_\upsilon|$. Focusing only on the factorization of $|\mathbf{L}_\upsilon|$ could also be seen by considering the tangent map relating the tangent space $\mathfrak{se}(\mathbf{3})$ to the tangent space of features $T_{\mathbf{s_{sp}}} = T_{\mathbf{s}_t} \times T_{{}^c\mathbf{R}_{p_{is}}} \mathbf{SO}(\mathbf{3})$ given by

$$\begin{bmatrix} \dot{\mathbf{s}}_t \\ \boldsymbol{\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_\upsilon & \mathbf{0} \\ \mathbf{L}_{\omega\upsilon} & -\mathbf{I_3} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}. \tag{47}$$

Indeed, the determinant of the tangent map (47) is given by $-|\mathbf{L}_\upsilon|$.

Using $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$, the block matrix $\mathbf{L}_\upsilon$ of the interaction matrix (18) can be rewritten as

$$\mathbf{L}_\upsilon = \begin{bmatrix} \alpha_{01}(k_{10}\mathbf{P}_0^\top + k_{01}\mathbf{P}_1^\top) \\ \alpha_{02}(k_{20}\mathbf{P}_0^\top + k_{02}\mathbf{P}_2^\top) \\ \alpha_{12}(k_{21}\mathbf{P}_1^\top + k_{12}\mathbf{P}_2^\top) \end{bmatrix} \tag{48}$$

where $\alpha_{ij} = 1/d_{ij}\|\mathbf{P}_i\|\|\mathbf{P}_j\|$, and $k_{ij} = 1 - (\|\mathbf{P}_i\|/\|\mathbf{P}_j\|) \cos(\mathbf{P}_i, \mathbf{P}_j)$.

The determinant of $\mathbf{L}_\upsilon$ can be easily computed from the determinant of its transpose $\mathbf{L}_\upsilon^\top$. Indeed, from the fact that $|\mathbf{L}_\upsilon| = \left|\mathbf{L}_\upsilon^\top\right|$, we have

$$|\mathbf{L}_\upsilon| = \left|\mathbf{L}_{\upsilon x}^\top, \mathbf{L}_{\upsilon y}^\top, \mathbf{L}_{\upsilon z}^\top\right| \tag{49}$$

where

$$\begin{cases} \mathbf{L}_{\upsilon x}^\top = \alpha_{01}(k_{10}\mathbf{P}_0 + k_{01}\mathbf{P}_1) \\ \mathbf{L}_{\upsilon y}^\top = \alpha_{02}(k_{20}\mathbf{P}_0 + k_{02}\mathbf{P}_2) \\ \mathbf{L}_{\upsilon z}^\top = \alpha_{12}(k_{21}\mathbf{P}_1 + k_{12}\mathbf{P}_2). \end{cases}$$

Using the multilinear property of the determinant application, from (49), we obtain after some developments,

$$|\mathbf{L}_\upsilon| = \alpha_{01}\alpha_{02}\alpha_{12}(k_{10}k_{02}k_{21} + k_{01}k_{20}k_{12})\,|\mathbf{P}_1, \mathbf{P}_0, \mathbf{P}_2| \tag{50}$$

where $\alpha_{01}\alpha_{02}\alpha_{12} \neq 0$ since $\alpha_{ij} \neq 0, i = 0, 1, j = 1, 2$. From (50), it is clear that, if the three points are aligned, then $|\mathbf{P}_1, \mathbf{P}_0, \mathbf{P}_2| = 0$.

Now, we show that in the case where the three points are not aligned, the expression

$$k_{10}k_{02}k_{21} + k_{01}k_{20}k_{12} = 0 \tag{51}$$

characterizes the cylinder of singularities [see Fig. 3] defined in [23]. Using the expressions of k_{ij} given in (48), it is possible to show after some developments that (51) is equivalent to

$$(\mathbf{P}_1\mathbf{P}_0^\top\mathbf{C}\mathbf{P}_0)(\mathbf{P}_0\mathbf{P}_2^\top\mathbf{C}\mathbf{P}_2)(\mathbf{P}_2\mathbf{P}_1^\top\mathbf{C}\mathbf{P}_1) \\ +(\mathbf{P}_0\mathbf{P}_1^\top\mathbf{C}\mathbf{P}_1)(\mathbf{P}_2\mathbf{P}_0^\top\mathbf{C}\mathbf{P}_0)(\mathbf{P}_1\mathbf{P}_2^\top\mathbf{C}\mathbf{P}_2) = 0 \tag{52}$$

Expression (52) is easily verified for the particular configuration of the cylinder of singularities illustrated in Fig. 3(a), where $(\mathbf{P}_2\mathbf{P}_1^\top\mathbf{C}\mathbf{P}_1) = (\mathbf{P}_0\mathbf{P}_1^\top\mathbf{C}\mathbf{P}_1) = 0$.

Let the point $\mathcal{P}$ be the orthogonal projection of the camera optical center $\mathcal{C}$ onto the plane defined by the three points $\mathcal{P}_0$, $\mathcal{P}_1$, and $\mathcal{P}_2$ [see Fig. 3(b)]. Let $\mathcal{F}_c'$ be a frame centered in $\mathcal{C}$ and oriented such that $\mathbf{z} = \mathbf{CP}/\|\mathbf{CP}\|$. In $\mathcal{F}_c'$, the coordinates of $\mathcal{P}$ are given by $(0, 0, P_z')$, and points $\mathcal{P}_0$, $\mathcal{P}_1$, and $\mathcal{P}_2$ have all the same z-component $P_{iz}' = P_z'$. We denote $\mathbf{P}_i' = (P_{ix}', P_{iy}', P_z')$ as the vector coordinates of $\mathcal{P}_i$ in $\mathcal{F}_c'$.

Equation (52) still holds in $\mathcal{F}_c'$ since rotation preserves dot product. By expressing (52) in $\mathcal{F}_c'$, expanding, and simplifying, we get the product of two determinants

$$\begin{vmatrix} P_{0x}' & P_{0y}' & 1 \\ P_{1x}' & P_{1y}' & 1 \\ P_{2x}' & P_{2y}' & 1 \end{vmatrix} \cdot \begin{vmatrix} P_{0x}' & P_{0y}' & P_{0x}'^2 + P_{0y}'^2 \\ P_{1x}' & P_{1y}' & P_{1x}'^2 + P_{1y}'^2 \\ P_{2x}' & P_{2y}' & P_{2x}'^2 + P_{2y}'^2 \end{vmatrix} = 0.$$

Since the three points are not aligned, we have, on one hand,

$$\begin{vmatrix} P_{0x}' & P_{0y}' & 1 \\ P_{1x}' & P_{1y}' & 1 \\ P_{2x}' & P_{2y}' & 1 \end{vmatrix} \neq 0.$$

On the other hand, we must have

$$\begin{vmatrix} P_{0x}' & P_{0y}' & P_{0x}'^2 + P_{0y}'^2 \\ P_{1x}' & P_{1y}' & P_{1x}'^2 + P_{1y}'^2 \\ P_{2x}' & P_{2y}' & P_{2x}'^2 + P_{2y}'^2 \end{vmatrix} = 0. \tag{53}$$

The camera optical center $\mathcal{C}$ belongs to the cylinder of singularities iff $\mathcal{P}$ belongs to the circumcircle defined by the three points. The property that the point $\mathcal{P}$ of coordinates $(0, 0, P_z')$ belongs to the circumcircle defined by the three points can be expressed by the following three-point formula for the circle [29]

$$\begin{vmatrix} 0 & 0 & 0 & 1 \\ P_{0x}'^2 + P_{0y}'^2 & P_{0x}' & P_{0y}' & 1 \\ P_{1x}'^2 + P_{1y}'^2 & P_{1x}' & P_{1y}' & 1 \\ P_{2x}'^2 + P_{2y}'^2 & P_{2x}' & P_{2y}' & 1 \end{vmatrix} = 0 \qquad (54)$$

which is clearly equivalent to expression (53). The cylinder of singularities described in Fig. 3 is thus characterized by the expression (53), which is equivalent to (51). ■

ACKNOWLEDGMENT

The authors gratefully acknowledge S. Hutchinson for his comments on earlier drafts of this paper, as well as the anonymous reviewers.

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Rob. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[2] *Visual Servoing via Advanced Numerical Methods*, G. Chesi and K. Hashimoto, Eds. New York: Springer-Verlag, 2010.

[3] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *Confluence Vision Control* (LNCIS, no. 237). New York: Springer-Verlag, 1998, pp. 66–78.

[4] R. Basri, E. Rivlin, and I. Shimshoni, "Visual homing: Surfing on the epipoles," *Int. J. Comput, Vision*, vol. 33, no. 2, pp. 117–137, Feb. 1999.

[5] D. Burschka and G. Hager, "Vision-based control of mobile robots," in *Proc. IEEE Int. Conf. Rob. Autom.*, 2001, pp. 1707–1713.

[6] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette, "Outdoor visual path following experiments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, Oct. 2007, pp. 4265–4270.

[7] T. Hamel and R. Mahony, "Visual servoing of an under-actuated dynamic rigid-body system: An image-based approach," *IEEE Trans. Rob. Autom.*, vol. 18, no. 2, pp. 187–198, Apr. 2002.

[8] S. Saripalli, J. Montgomery, and G. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Trans. Rob. Autom.*, vol. 19, no. 3, pp. 371–380, Jun. 2003.

[9] W. Wilson, C. Hulls, and G. Bell, "Relative end-effector control using cartesian position-based visual servoing," *IEEE Trans. Rob. Autom.*, vol. 12, no. 5, pp. 684–696, Oct. 1996.

[10] P. Martinet, J. Gallice, and D. Khadraoui, "Vision based control law using 3d visual features," in *Proc. World Autom. Congr.*, vol. 3, Montpellier, France, May 1996, pp. 497–502.

[11] G. Chesi and A. Vicino, "Visual servoing for large camera displacements," *IEEE Trans. Rob.*, vol. 20, no. 4, pp. 724–735, Aug. 2004.

[12] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Trans. Rob. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.

[13] N. Gans, A. Dani, and W. Dixon, "Visual servoing to an arbitrary pose with respect to an object given a single known length," in *Proc. Amer. Control Conf.*, 2008, pp. 4791–4796.

[14] J. T. Feddema, C. S. G. Lee, and O. R. Mitchell, "Automatic selection of image features for visual servoing of a robot manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Scottsdale, AZ, May 1989, pp. 832–837.

[15] M. Iwatsuki and N. Okiyama, "A new formulation for visual servoing based on cylindrical coordinate system," *IEEE Trans. Rob.*, vol. 21, no. 2, pp. 266–273, Apr. 2005.

[16] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Rob. Autom.*, vol. 17, no. 4, pp. 507–515, Aug. 2001.

[17] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Trans. Rob.*, vol. 21, no. 6, pp. 1116–1127, Dec. 2005.

[18] O. Tahri, Y. Mezouar, F. Chaumette, and P. Corke, "Decoupled image-based visual servoing for cameras obeying the unified projection model," *IEEE Trans. Rob.*, vol. 26, no. 4, pp. 684–697, Aug. 2010.

[19] R. Tatsambon Fomena and F. Chaumette, "Improvements on visual servoing from spherical targets using a spherical projection model," *IEEE Trans. Rob.*, vol. 25, no. 4, pp. 874–886, Aug. 2009.

[20] R. Tatsambon Fomena, O. Tahri, and F. Chaumette, "Visual servoing from three points using a spherical projection model," in *Proc. IEEE Int. Conf. Rob. Autom.*, Anchorage, AK, May 2010.

[21] R. P. Horaud, "New methods for matching 3d objects with single perspective views," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 3, pp. 401–412, May 1987.

[22] N. Cowan and D. Chang, "Geometric visual servoing," *IEEE Trans. Rob.*, vol. 21, no. 6, pp. 1128–1138, Dec. 2005.

[23] H. Michel and P. Rives, "Singularities in the determination of the situation of a robot," INRIA, France, Tech. Rep. 1850, Feb. 1993.

[24] A. De Luca, G. Oriolo, and P. R. Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *Int. J. Rob. Res.*, vol. 27, no. 10, pp. 1093–1116, 2008.

[25] F. Chaumette, P. Rives, and B. Espiau, "Classification and realization of the different vision-based tasks," in *Visual Servoing* (Series in Robotics and Automated Systems, vol. 7). Singapore: World Scientific, 1993, pp. 199–228.

[26] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," in *Proc. Eur. Conf. Comput. Vision*, 2000, pp. 445–461.

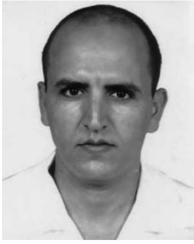
[27] E. Malis, "Contributions à la modélisation et à la commande en asservissement visuel," Ph.D. dissertation, Univ. de Rennes 1, Rennes, France, Nov. 1998.

[28] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: A generic software platform with a wide class of robot control skills," *IEEE Rob. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, Dec. 2005.

[29] H. S. M. Coxeter, *Introduction to Geometry*. New York: Wiley, 1969.

Romeo Tatsambon Fomena (M'09) received the Bachelor degree in mathematics and computer science from the University of Ngaoundere, Ngaoundere, Cameroon, in 2002 and the Masters and Ph.D. degrees in computer science from the Université de Rennes 1, Rennes, France, in 2005 and 2008, respectively.

For one month, he was with the University of Illinois at Urbana-Champaign in the framework of a collaborative work between the Beckman Institute and INRIA in 2008. He is currently a Postdoctoral Research Associate with Cemagref, Rennes, France, which is the French Environmental Sciences and Technologies Research Institute. His current research interests include vision-based control of robots and fluid flows control.



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

Since 2008, he has been a Researcher with the Computer Vision Laboratory, Institute for Systems and Robotics, Coimbra, Portugal. His current research interests include robotics and computer vision, especially visual servoing

François Chaumette (SM'10) received the Graduate degree from the École Nationale Supérieure de Mécanique, Nantes, France, in 1987 and the Ph.D. degree in computer science from the University of Rennes, Rennes, France, in 1990.

He is currently the Head of the Lagadic Group, INRIA, Centre Rennes-Bretagne Atlantique, and IRISA, Rennes, France. His current research interests include robotics and computer vision, especially visual servoing and active perception.