Efficient Iterative Pose Estimation using an Invariant to Rotations

Omar Tahri, Helder Araujo, Youcef Mezouar, François Chaumette

▶ **To cite this version:**

Omar Tahri, Helder Araujo, Youcef Mezouar, François Chaumette. Efficient Iterative Pose Estimation using an Invariant to Rotations. IEEE Transactions on Cybernetics, 2014, 22 (4), pp.199-207. 10.1109/TCYB.2013.2251331 . hal-00840264

HAL Id: hal-00840264

https://inria.hal.science/hal-00840264

Submitted on 2 Jul 2013

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

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

Efficient Iterative Pose Estimation using an Invariant to Rotations

Omar Tahri, Helder Araujo,Youcef Mezouar and François Chaumette

***Résumé*—This paper deals with pose estimation using an iterative scheme. We show that using adequate visual information, pose estimation can be performed iteratively with only three independent unknowns, which are the translation parameters. Specifically, an invariant to rotational motion is used to estimate the camera position. In addition, an adequate transformation is applied to the proposed invariant to decrease the non-linearities between the variations in image space and 3D space. Once the camera position is estimated, we show that the rotation can be estimated efficiently using two different direct methods. The proposed approach is compared against two other methods from the literature. The results show that using our method, pose tracking in image sequences and the convergence rate for randomly generated poses are improved.**

***Index Terms*—Pose estimation, invariant to rotation, projection onto sphere**

I. Introduction

Pose estimation (also known as extrinsic camera calibration) consists on the determination of the position and orientation of a camera with respect to an object coordinate frame using the image information. Pose estimation is a classical problem in computer vision [6], [15]. Nevertheless, there is a recent renewed interest as a result of automated navigation and model-based vision systems.

Numerous methods to estimate the pose have been proposed in the literature and giving an exhaustive list of them is certainly impossible. Nevertheless, they can be divided into several categories according to the features used or the nature of the estimation method, namely direct methods or iterative methods. The geometric features considered for the estimation of the pose are often points [6], segments [7], contours, conics [18] or image moments [22]. Another important issue is the registration problem. Purely geometric [7], or numerical and iterative [6], [2], [16] approaches may be considered. Linear approaches give closed-form solutions free of initialization [8], [1], [14]. However, the estimated pose using such methods is sensitive to image noise and to errors on camera parameters. Full-scale non-linear optimization techniques [16] minimize the error between the observation and the projection of the feature using the model, that is the reprojection error. The main advantage of the non-linear and iterative approaches is their accuracy. The main drawback is that they may be subject to local minima and, worse, divergence, if not correctly initialized. Furthermore, they usually require several iterations to minimize the cost function and generally they are more time consuming than the direct methods. These problems (i.e. local minima, divergence and time cost) are mainly due to non-linearities in the mapping between 3D and image space. The non-linearities are also usually the main reason for the failure of filtering strategies of the pose [13]. This occurs especially when the initial state is not accurate or when abrupt motions happen (for instance, for Extended Kalman Filter [21]).

In this paper, we deal with the selection of visual information that decreases the effect of the non-linearities between the variations in the image space and the 3D space. Firstly, we will show that the iterative estimation of the pose can be expressed as a minimization without constraints of a cost function on three unknowns only, which are the translation parameters. More precisely, to estimate the camera position separately from the rotation, an invariant to rotation computed from the point projections onto the unit sphere will be used. Furthermore, these visual features are chosen to minimize the changes on their corresponding Jacobian matrices with respect to the camera position. Secondly, we will also show that once the camera position is obtained using an iterative method, the rotation can be computed directly, that is, it is obtained without any iterative method. Therefore, the convergence speed and rate are only function of the translations.

In the next section, we recall the unified projection model and some basic definitions. Section 3 describes

O. Tahri and H. Araujo are with ISR, University of Coimbra, Polo II PT-3030-290 Coimbra, Portugal (omartahri@isr.uc.pt, helder@isr.uc.pt).

Y. Mezouar is with IFMA - Institut Français de Mécanique Avancée, Institut Pascal, BP 10448, F-63000 Clermont-Ferrand, France (youcef.mezouar@ifma.fr).

F. Chaumette is with Inria Rennes-Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes-cedex, France (Francois.Chaumette@irisa.fr).

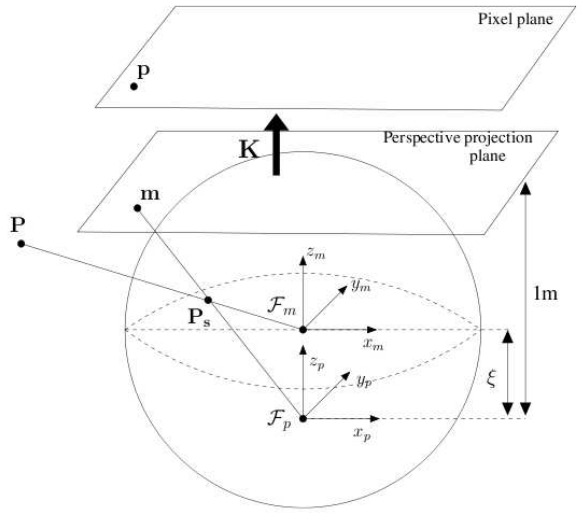


FIGURE 1. Unified image formation.

the method we propose. Section 4 compares this method with two iterative methods from the literature.

II. DEFINITIONS

A. Notations

In the sequel, the following notations will be used :

- $\mathbf{P} = (X, Y, Z)$: 3D point coordinates ;
- $\mathbf{P_s} = (x_s, y_s, z_s)$: projected point onto the unit sphere ;
- $\mathbf{P_v}$: virtual points defined by linear combination of the projected points onto the unit sphere ;
- $\mathbf{m} = (x, y, 1)$: coordinates of projected point onto the image plane in metric units ;
- $\mathbf{p}$: coordinates of projected point onto the image plane in pixels ;
- $d_{ij} = \sqrt{2 - 2\mathbf{P}_{s_i}^\top \mathbf{P}_{s_j}}$: distance between two projected points $\mathbf{P}_{s_i}$ and $\mathbf{P}_{s_j}$ on the unit sphere ;
- the variables followed by $*$ are computed for the camera pose to be estimated.
- all the scalars are in italic.
- all the matrices and vectors are in bold

B. Camera Model

Central imaging systems can be modeled using two consecutive projections : spherical and then perspective. This geometric formulation, called the *unified model*, was proposed by Geyer and Daniilidis in [9]. Consider $\mathcal{F}_m$ the frame attached to a virtual unitary sphere as shown on Fig. 1. The frames attached to the sphere $\mathcal{F}_m$ and to the perspective camera $\mathcal{F}_p$ are related by a simple translation of $-\xi$ along the Z-axis. Let $\mathbf{P}$ be a 3D point with coordinates $\mathbf{P} = (X, Y, Z)$ in $\mathcal{F}_m$. The world point $\mathbf{P}$ is projected onto :

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

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 be obtained from calibration using, for example, the methods proposed in [17]. In the sequel, the imaging system is assumed to be calibrated. In this case, the inverse projection onto the unit sphere can be obtained from :

$$\mathbf{P_s} = \gamma \begin{pmatrix} x, & y, & 1 - \frac{\xi}{\gamma} \end{pmatrix} \quad (2)$$

where

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

The projection onto the unit sphere from the image plane is possible for all sensors obeying the unified model. In other words, it encompasses all sensors in this class [9] : perspective and catadioptric cameras. A large class of fisheye cameras are also concerned by this model [4], [3].

C. Pose estimation

Pose estimation consists in determining the rigid transformation ${}^c\mathbf{M}_o$ between the object frame $\mathcal{F}_o$ and the camera frame $\mathcal{F}_c$ in unknown position using the corresponding object image. It is well known that the relationship between an object point with coordinates $\mathbf{P}_c = [X_c,\ Y_c,\ Z_c,\ 1]^\top$ in $\mathcal{F}_c$ and $\mathbf{P}_o = [X_o,\ Y_o,\ Z_o,\ 1]^\top$ in $\mathcal{F}_o$ can be written :

$$\mathbf{P}_c = {}^c\mathbf{M}_o\, \mathbf{P}_o = \begin{bmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ \mathbf{0_{31}} & 1 \end{bmatrix} \mathbf{P}_o. \quad (3)$$

The matrix ${}^c\mathbf{M}_o$ can be estimated by minimizing the modulus of the error in the image :

$$e = \| \mathbf{s}({}^c\mathbf{M}_o) - \mathbf{s}^* \|, \quad (4)$$

where $\mathbf{s}^*$ is the value of a set of visual features computed in the image acquired with the camera in unknown position and $\mathbf{s}({}^\mathbf{c}\mathbf{M_o})$ is the value of the same set of features computed from the object model, the transformation ${}^c\mathbf{M}_o$, and the camera model.

III. POSE ESTIMATION METHOD

In this section, we first propose new features to estimate the camera position separately from the rotation. We then propose a method for the direct estimation of the rotation once the translational part of the pose has been determined.

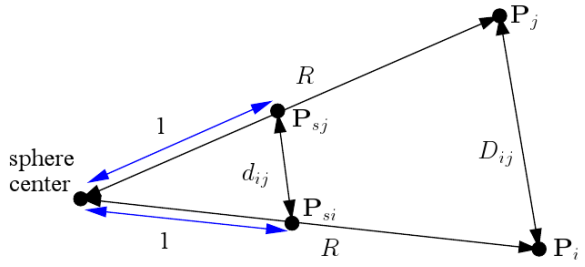


FIGURE 2. Relation between 3D distance and distance between projected point on the sphere

A. Position estimation using an invariant to rotation

1) Invariant to rotations: Let d_{ij} be the distance between two projected points on the unit sphere

$$d_{ij} = \sqrt{2 - 2\,\mathbf{P}_{s_i}^{\top}\mathbf{P}_{s_j}} \tag{5}$$

It can easily be shown that the distance d_{ij} is an invariant to any rotational motion applied to the camera frame. Indeed, if we apply a rotational motion to the camera frame, the projected points on the sphere undergo also the same rotation. In this case, we have $\mathbf{P}'_{s_i} = \mathbf{R}\mathbf{P}_{s_i}$, where $\mathbf{P}'_{s_i}$ and $\mathbf{P}_{s_i}$ are the projections of a 3D point onto the unit sphere corresponding the two cameras orientations and $\mathbf{R}$ is the matrix defining the rotation. Therefore, it can be obtained that $\mathbf{P}'^{\top}_{s_i}\mathbf{P}'_{s_j} = \mathbf{P}_{s_i}\,\mathbf{R}^{\top}\,\mathbf{R}\,\mathbf{P}_{s_j} = \mathbf{P}^{\top}_{s_i}\mathbf{P}_{s_j}$ since $\mathbf{R}^{\top}\,\mathbf{R} = \mathbf{I}$, which proves the invariance of d_{ij} to rotational motion applied to the camera frame. Therefore, the variation of d_{ij} only depends of the translation. Furthermore, the Jacobian matrix that links the variation of d_{ij} with respect to translational displacement is given by :

$$\mathbf{J}_{d_{ij}} = -\frac{\mathbf{P}^{\top}_{s_i}\mathbf{J}_{\mathbf{P}_{s_j}} + \mathbf{P}^{\top}_{s_j}\mathbf{J}_{\mathbf{P}_{s_i}}}{d_{ij}} \tag{6}$$

where $\mathbf{J}_{\mathbf{P}_{s_i}}$ and $\mathbf{J}_{\mathbf{P}_{s_j}}$ are the Jacobian matrices that relate the variations of the point coordinates on the unit sphere to the camera translational displacements. This Jacobian has the following form [11] :

$$\mathbf{J}_{\mathbf{P}_{s_i}} = \frac{-\mathbf{I} + \mathbf{P}_{s_i}\mathbf{P}^{\top}_{s_i}}{\parallel \mathbf{P_i} \parallel} \tag{7}$$

where $\parallel \mathbf{P_i} \parallel$ is the distance of the 3D point to the center of the sphere. After inserting (7) in (6), we obtain :

$$\begin{aligned}\mathbf{J}_{d_{ij}} = -\frac{1}{d_{ij}}\Bigg(&(-\frac{1}{\parallel \mathbf{P_j} \parallel} + \frac{\mathbf{P}^{\top}_{s_i}\mathbf{P}_{s_j}}{\parallel \mathbf{P_i} \parallel})\mathbf{P}^{\top}_{s_i} \\ &+(-\frac{1}{\parallel \mathbf{P_i} \parallel} + \frac{\mathbf{P}^{\top}_{s_i}\mathbf{P}_{s_j}}{\parallel \mathbf{P_j} \parallel})\mathbf{P}^{\top}_{s_j}\Bigg)\end{aligned} \tag{8}$$

Further to the invariance to rotation, which allows separating the estimation of the camera position and orientation, it is also possible to decrease the non-linearities between the image space and 3D space. Indeed, the distance d_{ij} on the sphere behaves as function which is approximately inversely proportional to the point depths $\parallel \mathbf{P_i} \parallel$. This means that its corresponding Jacobian matrix depends on the square of the inverse of the point depths. On the other hand, the inverse of the distance behaves approximately as a linear function of the points depths. This should allow obtaining more linearizing properties between the image space and 3D space. So we propose to use $s_{ij} = 1/d_{ij}$ for all possible combinations of two projected points. Let us consider the case when the "mean" distance R of the points to the sphere center is such that $R \approx \parallel \mathbf{P_i} \parallel \approx \parallel \mathbf{P_j} \parallel$ as shown in Figure 2. In this case, we have :

$$\mathbf{J}_{s_{ij}} \approx \frac{(-1 + \mathbf{P}^{\top}_{s_i}\mathbf{P}_{s_j})}{R}\frac{(\mathbf{P}_{s_i} + \mathbf{P}_{s_j})^{\top}}{d_{ij}^3} \tag{9}$$

Note that $-1+\mathbf{P}^{\top}_{s_i}\mathbf{P}_{s_j} = -\frac{d_{ij}^2}{2}$, then (9) can be written as :

$$\mathbf{J}_{s_{ij}} \approx \frac{\parallel \mathbf{P}_{s_i} + \mathbf{P}_{s_j} \parallel}{2R\,d_{ij}}\frac{(\mathbf{P}_{s_i} + \mathbf{P}_{s_j})^{\top}}{\parallel \mathbf{P}_{s_i} + \mathbf{P}_{s_j} \parallel} \tag{10}$$

From Figure 2, we have :

$$\frac{d_{ij}}{D_{ij}} = \frac{1}{R} \tag{11}$$

By combining (11) with (10), we obtain :

$$\mathbf{J}_{s_{ij}} \approx \frac{\parallel \mathbf{P}_{s_i} + \mathbf{P}_{s_j} \parallel}{2D_{ij}}\frac{(\mathbf{P}_{s_i} + \mathbf{P}_{s_j})^{\top}}{\parallel \mathbf{P}_{s_i} + \mathbf{P}_{s_j} \parallel} \tag{12}$$

Note that $\frac{(\mathbf{P}_{s_i}+\mathbf{P}_{s_j})^{\top}}{\|\mathbf{P}_{s_i}+\mathbf{P}_{s_j}\|}$ is the unitary vector that passes through the middle of the two points $\mathbf{P}_{s_i}$ and $\mathbf{P}_{s_j}$ and also $\parallel \mathbf{P}_{s_i}+\mathbf{P}_{s_j} \parallel \approx 2$ if $R >> D_{ij}$. This means that the Jacobian matrix $\mathbf{J}_{s_{ij}}$ behaves as a constant matrix when the point depth increases.

2) Noise propagation from image space to the new feature space: Applying non-linear transformations on data obtained from the sensor space changes the noise distribution. For instance, if the image noise is Gaussian white, the noise on invariants features is no more Gaussian white since the applied transformation is nonlinear. In practice, if the noise level is low, the use of invariants s_{ij} allows obtaining adequate performances. If the noise level increases, the propagation of noise from the image to the feature s_{ij} should be taken into account. Let us start with the sensitivity of a projected point onto the sphere with respect to noise in the image plane. Taking the derivative of (2), the variation in the coordinates of

the point projected onto the sphere as a function of the variation in the coordinates in the image points (noise-meters) is obtained by (using first order approximation) :

$$\Delta \mathbf{P}_s = \mathbf{J}_{\mathbf{P}_s/\mathbf{m}} \Delta \mathbf{m} \tag{13}$$

where :

$$\mathbf{J}_{\mathbf{P}_s/\mathbf{m}} = \begin{bmatrix} \gamma + x\frac{\partial \gamma}{\partial x} & x\frac{\partial \gamma}{\partial y} & 0 \\ y\frac{\partial \gamma}{\partial x} & \gamma + y\frac{\partial \gamma}{\partial y} & 0 \\ \frac{\partial \gamma}{\partial x} & \frac{\partial \gamma}{\partial y} & 0 \end{bmatrix} \tag{14}$$

with :

$$\begin{aligned} \frac{\partial \gamma}{\partial x} &= \frac{x}{1+x^2+y^2}\left(\frac{(1-\xi^2)}{\sqrt{1+(1-\xi^2)(x^2+y^2)}} - 2\gamma\right) \\ \frac{\partial \gamma}{\partial y} &= \frac{y}{1+x^2+y^2}\left(\frac{(1-\xi^2)}{\sqrt{1+(1-\xi^2)(x^2+y^2)}} - 2\gamma\right) \end{aligned} \tag{15}$$

where γ and ξ have been defined in Section II-B. Therefore, the variation of $\mathbf{P}_s$ with respect to image points in pixels is obtained by :

$$\Delta \mathbf{P}_s = \mathbf{J}_{\mathbf{P}_s/\mathbf{m}} \mathbf{K}^{-1} \Delta \mathbf{p} \tag{16}$$

Furthermore, from $d_{ij} = \sqrt{2 - 2\mathbf{P}_{si}^\top \mathbf{P}_{sj}}$, we have :

$$\Delta d_{ij} = -\frac{1}{d_{ij}} (\mathbf{P}_{sj}^\top \Delta \mathbf{P}_{si} + \mathbf{P}_{si}^\top \Delta \mathbf{P}_{sj}) \tag{17}$$

As a result of (14) and (17), the variation of $s_{ij} = \frac{1}{d_{ij}}$ with respect to noise in the coordinates of the image points (in pixels) is obtained by :

$$\Delta s_{ij} = \mathbf{J}_{s_{ij}/\mathbf{p}} \begin{bmatrix} \Delta \mathbf{p}_i \\ \Delta \mathbf{p}_j \end{bmatrix} \tag{18}$$

where $\mathbf{J}_{s_{ij}/\mathbf{p}} = \begin{bmatrix} \mathbf{P}_{s_j}^\top \ \mathbf{J}_{\mathbf{P}_{si}/\mathbf{m_i}} \mathbf{K}^{-1} & \mathbf{P}_{s_i}^\top \ \mathbf{J}_{\mathbf{P}_{sj}/\mathbf{m_j}} \mathbf{K}^{-1} \end{bmatrix} / d_{ij}^3$. In order to take into account the non-linear mapping from the image point coordinates to the features s_{ij}, each visual feature should be weighted by $\frac{1}{\|\mathbf{J}_{s_{ij}^*/\mathbf{p}^*}\|}$ computed using the image points coordinates corresponding to the pose to be computed. More precisely, we use all possible combinations of $s_{wij} = \frac{1}{d_{ij}} \frac{1}{\|\mathbf{J}_{s_{ij}^*/\mathbf{p}^*}\|}$ as measure to estimate the camera position.

3) Direct estimation of the rotation: We now present two direct methods to determine the rotation once the translation has been estimated.

a) Method based on an orthonormal basis computation: In [5] and [23], an angle-axis representation of a rotation matrix $\mathbf{R}$ computed from two projected points on the sphere has been considered to control rotational motions for visual servoing application. The idea behind the rotation formula given in these works is equivalent to attaching an orthonormal frame basis to each camera pose using two projected points onto the unit sphere. More precisely, let $\mathbf{P}_{s_1}$ and $\mathbf{P}_{s_2}$ be two projected points on the sphere for a pose 1 of the camera and ${\mathbf{P}^*}_{s_1}$ and $\mathbf{P}^*_{s_2}$ their corresponding projected points for a camera pose 2. From $\mathbf{P}_{s_1}$ and $\mathbf{P}_{s_2}$, it is possible to define an orthonormal basis ${}^c\mathbf{R}_s = [\mathbf{v}_1; \mathbf{v}_2; \mathbf{v}_3]$ (see Figure 3) such that :

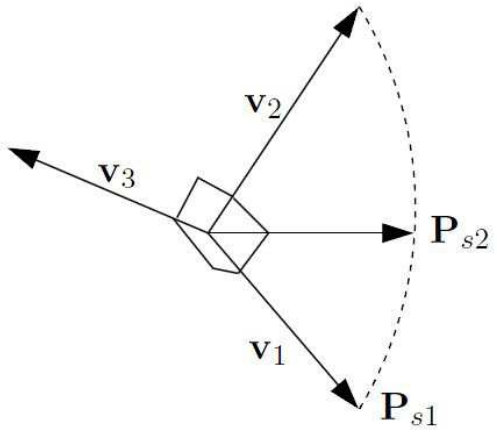


FIGURE 3. Definition of vector basis from 2 projected points

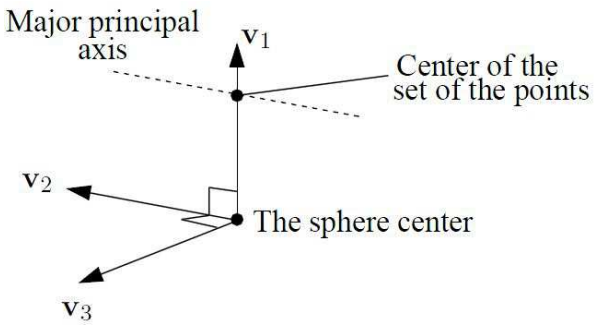


FIGURE 4. Definition of vector basis from N projected points

$$\mathbf{v}_1 = \mathbf{P}_{s_1}, \tag{19}$$

$$\mathbf{v}_2 = \frac{\mathbf{P}_{s_2} - (\mathbf{P}_{s_2}^\top \mathbf{P}_{s_1})\mathbf{P}_{s_1}}{\| \mathbf{P}_{s_2} - (\mathbf{P}_{s_2}^\top \mathbf{P}_{s_1})\mathbf{P}_{s_1} \|}, \tag{20}$$

$$\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2. \tag{21}$$

and similarly an orthonormal basis ${}^{c*}\mathbf{R}_{s*}$ using points ${\mathbf{P}^*}_{s_1}$ and $\mathbf{P}^*_{s_2}$. If only a rotational motion is considered, the rotation matrix $\mathbf{R}$ between the two camera poses is determined by the matrix that transforms the vector basis ${}^c\mathbf{R}_s$ to ${}^{c*}\mathbf{R}_{s*}$:

$$\mathbf{R} = {}^{c*}\mathbf{R}_{s*} \ {}^c\mathbf{R}_s^\top \tag{22}$$

For the sake of robustness, all projected points on the sphere should be used and not only two. In this work,

we propose a way to define the rotation matrix using all the points. The idea is based on the fact that the rotation matrix given by (22) can be obtained from two real projected points as well as from two virtual points rigidly attached to the set of projected points. Let us consider :

$$\mathbf{P}_{v_1} = \sum_{i=1}^{N} a_{1i}\mathbf{P}_{s_i}, \quad \mathbf{P}_{v_2} = \sum_{i=1}^{N} a_{2i}\mathbf{P}_{s_i} \tag{23}$$

two virtual points obtained by a linear combination of the real set of projected points on the sphere. Then, from $\mathbf{P}_{v_1}$ and $\mathbf{P}_{v_2}$ an orthonormal basis ${}^c\mathbf{R}_n$ can be defined as follows :

$$\mathbf{v}_{n1} = \frac{\mathbf{P}_{v_1}}{\| \mathbf{P}_{v_1} \|}, \tag{24}$$

$$\mathbf{v}_{n2} = \frac{\mathbf{P}_{v_2} - \mathbf{P}_{v_2}^\top \mathbf{v}_{n1}\mathbf{v}_{n1}}{\| \mathbf{P}_{v_2} - \mathbf{P}_{v_2}^\top \mathbf{v}_{n1}\mathbf{v}_{n1} \|}, \tag{25}$$

$$\mathbf{v}_{n3} = \mathbf{v}_{n1} \times \mathbf{v}_{n2}. \tag{26}$$

Lemma 1: If only a rotational motion is considered, the rotation matrix $\mathbf{R}$ between the two camera poses is determined by the matrix that transforms the vector basis ${}^c\mathbf{R}_n$ into ${}^{c*}\mathbf{R}_{n*}$:

$$\mathbf{R} = {}^{c*}\mathbf{R}_{n*}\ {}^c\mathbf{R}_n^\top \tag{27}$$

where :

$${}^c\mathbf{R_n} = [\mathbf{v}_{n1};\ \mathbf{v}_{n2};\ \mathbf{v}_{n3}]$$

and

$${}^{c*}\mathbf{R_{n*}} = [\mathbf{v}^*_{n1};\ \mathbf{v}^*_{n2};\ \mathbf{v}^*_{n3}]$$

are computed using (23), (24), (25) and (26).
The proof of the previous lemma is detailed in the appendix.

To define the vector basis ${}^c\mathbf{R_n}$ and ${}^{c*}\mathbf{R}_{n*}$ it is necessary to determine the parameters a_{1i} and a_{2i} in (23). More precisely, we have to define two virtual points $\mathbf{P}^*_{v_1}$ and $\mathbf{P}^*_{v_2}$ and then express them as linear combinations of the projected points on the sphere $\mathbf{P}^*_{si}$ computed for the camera pose to be estimated. For the sake of simplicity, $\mathbf{P}^*_{v_1}$ and $\mathbf{P}^*_{v_2}$ are chosen to be unitary and perpendicular to each-other. For such case, the basis $[\mathbf{P}^*_{v_1};\ \mathbf{P}^*_{v_2};\ \mathbf{P}^*_{v_1} \times \mathbf{P}^*_{v_2}]$ is orthonormal.

In any given frame basis $[\mathbf{P}^*_{v_1};\ \mathbf{P}^*_{v_2};\ \mathbf{P}^*_{v_1} \times \mathbf{P}^*_{v_2}]$, each projected point onto the sphere can be expressed as :

$$\mathbf{P}^*_{si} = b_{1i}\mathbf{P}^*_{v_1} + b_{2i}\mathbf{P}^*_{v_2} + b_{3i}\mathbf{P}^*_{v_1} \times \mathbf{P}^*_{v_2} \tag{28}$$

Let $\mathbf{B}$ be the $3 \times N$ matrix that defines the coordinates of all the projected points on the new frame basis. We have :

$$\mathbf{P}^*_{st} = \mathbf{P}_{v*}\ \mathbf{B}, \tag{29}$$

where :

$$\mathbf{P}^*_{st} = [\mathbf{P}^*_{s1}\ \ \mathbf{P}^*_{s2} \dots \mathbf{P}^*_{sN}],$$

$$\mathbf{P}^*_{v} = [\mathbf{P}^*_{v_1}\ \mathbf{P}^*_{v_2}\ \mathbf{P}^*_{v_1} \times \mathbf{P}^*_{v_2}],$$

and

$$\mathbf{B} = \mathbf{P}^\top_{v*}\ \mathbf{P}^*_{st} \tag{30}$$

In practice, a_{1i} and a_{2i} have to be chosen such that their corresponding virtual points are robust to noise. Our choice is based on characteristic features of 3D structure—the center of gravity of the directions defined by the points and the principal axis of the directions (also defined by the points). More precisely, the first virtual point is defined by :

$$\mathbf{P}^*_{v_1} = \frac{1}{\| \sum_{i=1}^{N} \mathbf{P}^*_{si} \|} \sum_{i=1}^{N} \mathbf{P}^*_{si}. \tag{31}$$

which corresponds to $a_{1i} = \frac{1}{\|\sum_{i=1}^{N} \mathbf{P}^*_{si}\|}$. The second virtual point $\mathbf{P}^*_{v_2}$ is chosen as the unitary vector perpendicular to $\mathbf{P}^*_{v_1}$ that lays on the plane defined by $\mathbf{P}^*_{v_1}$ and the major principal axis of the set of the projected points on the sphere (see Fig. 4). The choice of the main principal axis as second axis allows having the majority of the points in its direction. Now as $\mathbf{P}^*_{v_1}$ and $\mathbf{P}^*_{v_2}$ have been determined, the matrix $\mathbf{B}$ can be computed using (30). From (28), it can be obtained that :

$$\mathbf{P}^*_{si} - b_{1i}\mathbf{P}^*_{v_1} = b_{2i}\mathbf{P}^*_{v_2} + b_{3i}\mathbf{P}^*_{v_1} \times \mathbf{P}^*_{v_2} \tag{32}$$

By replacing $\mathbf{P}^*_{v_1}$ by $\frac{\sum_{i=1}^{N} \mathbf{P}^*_{si}}{\|\sum_{i=1}^{N} \mathbf{P}^*_{si}\|}$, we obtain for all the projected points onto the sphere :

$$\mathbf{P}^*_{st}\mathbf{C} = \left[\ \mathbf{P}^*_{v_2} \quad \mathbf{P}^*_{v_1} \times \mathbf{P}^*_{v_2}\ \right] \mathbf{B_{23}} \tag{33}$$

where $\mathbf{B_{23}}$ is composed by the two last rows of $\mathbf{B}$ and $\mathbf{C}$ is an $N \times N$ matrix defined by $c_{lm} = 1 - \frac{b_{1l}}{\|\sum_{i=1}^{N} \mathbf{P}^*_{si}\|}$ if $l = m$ and $c_{lm} = -\frac{b_{1l}}{\|\sum_{i=1}^{N} \mathbf{P}^*_{si}\|}$ for $l \neq m$. By inverting (33), we obtain :

$$\left[\ \mathbf{P}^*_{v_2} \quad \mathbf{P}^*_{v_1} \times \mathbf{P}^*_{v_2}\ \right] = \mathbf{P}^*_{st}\,\mathbf{C}\,{\mathbf{B_{23}}}^{+}. \tag{34}$$

The parameters a_{2i} are then obtained as the first column of the matrix $\mathbf{C}\,{\mathbf{B_{23}}}^{+}$. Now as the coefficients a_{1i} and a_{2i} are defined, the vector basis for each camera pose can be obtained using (24), (25) and (26).

This way of obtaining the rotation matrix by building an orthogonal basis from a set of points has dual application in practice : it allows a direct estimation of rotation for pose estimation problem, but it can also provide a rotation vector from a set of N projected points on the

sphere to control the rotational motions in image-based visual servoing. In the next paragraph, we recall another direct way of recovering the rotation matrix between two set of projected points on the sphere.

b) Direct estimation of the rotation by solving an orthogonal Procrustes problem: The rotation matrix could also be directly obtained by solving orthogonal Procrustes problem between two sets of projected points on the sphere. We recall that the Orthogonal Procrustes problem is defined as the least square problem transforming a given matrix $\mathbf{F}$ into a given matrix $\mathbf{F'}$ by an orthogonal transformation $\mathbf{R}$ so that the sum of squares of the residual matrix $\mathbf{E} = \mathbf{RF} - \mathbf{F'}$ is minimal [12]. Mathematically, this problem can formulated as follows [19].

$$\mathbf{RF} = \mathbf{E} + \mathbf{F'}, \tag{35}$$

$$\mathbf{RR}^\top = \mathbf{I}, \tag{36}$$

$$tr(\mathbf{E}^\top \mathbf{E}) \; is \; minimal, \tag{37}$$

In our context, the matrices $\mathbf{F}$ and $\mathbf{F'}$ are composed by the set of all projected points onto unit sphere that is $\mathbf{P}_{st}$ and $\mathbf{P}^*_{st}$. The Orthogonal Procrustes Problem can be solved by computing the SVD decomposition of $\mathbf{P}^*_{st}\,\mathbf{P}^\top_{st}$:

$$\mathbf{P}^*_{st}\,\mathbf{P}^\top_{st} = \mathbf{U\Sigma V}^\top \tag{38}$$

The rotation matrix between the two camera poses is then given by :

$$\mathbf{R} = \mathbf{UV}^\top \tag{39}$$

4) Pose estimation algorithm: The pose estimation method is divided into two steps : firstly, we determine the translation between the initial pose and the pose to be estimated using the invariant to rotation as feature as follows :

- Project the image points corresponding to the pose to be computed onto the sphere using (2).
- Compute the value of features vector $\mathbf{s_t}^*$ for the pose to be estimated by stacking the features $s^*_{wij} = \frac{1}{d^*_{ij}} \frac{1}{\|\mathbf{J}_{s^*_{ij}/\mathbf{p}^*}\|}$.
- The camera pose is set up at its initial value :

$${}^c\mathbf{M}_o = \; {}^i\mathbf{M}_o = \begin{bmatrix} {}^i\mathbf{R}_o & {}^i\mathbf{t}_o \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$$

Minimization loop : while ($\| \mathbf{s_t} - \mathbf{s_t}^* \| \leq \epsilon$) where ϵ is defined by the user.

- Project the 3D points of the object onto the unit sphere using the object model and the current value of the pose ${}^c\mathbf{M}_o$.
- Compute the current value of features vector $\mathbf{s_t}$ corresponding to $\mathbf{s_t}^*$ by stacking the features $s_{wij} = \frac{1}{d_{ij}} \frac{1}{\|\mathbf{J}_{s^*_{ij}/\mathbf{p}^*}\|}$.
- Compute the Jacobian matrix $\mathbf{J}_{\mathbf{s_t}}$ corresponding to $\mathbf{s_t}$ ($\mathbf{J}_{\mathbf{s_t}}$ is an $l \times 3$ matrix, l is the number of used distances between projected points on the sphere).
- Compute the translational displacement using $\Delta\mathbf{t} = -\lambda \mathbf{J}^+_{\mathbf{s_t}}(\mathbf{s_t} - \mathbf{s_t}^*)$ (λ is a scalar gain that tunes the convergence speed and $\mathbf{J}^+_{\mathbf{s_t}}$ is the pseudo-inverse of $\mathbf{J}_{\mathbf{s_t}}$)
- Update ${}^c\mathbf{M}_o$ by adding the translational motion $\Delta\mathbf{t}$.

After the end of the minimization loop described above, the matrix ${}^c\mathbf{R}_i$ that defines the rotation between the initial camera pose (defined by ${}^i\mathbf{M}_o$) and the camera pose to be computed can be directly obtained using one of the two methods presented in Section III-A3. Besides, the rotation between the object frame and the camera frame is then obtained by ${}^c\mathbf{R}_o = {}^c\mathbf{R}_i \; {}^i\mathbf{R}_o$. This means that if the translational motion is well estimated using an invariant to rotations, the correct pose will be obtained. Let us remind that the iterative minimization process is nothing but a minimization without constraints of a cost function on three unknowns only. Therefore, the convergence speed and rate are only function of the translations.

IV. Validation results

In this part, our pose estimation method is compared to two non-linear and iterative methods proposed respectively by [2] (method A in the following) and by [16] (method L in the following). The method L is a globally convergent algorithm that minimizes error in object space : the error between the observation and the projection of the features using the model. On the other hand, the method A minimizes an error defined in the image and improves the classical Lowe's pose-estimation algorithm. A comparison of several iterative methods has been made in [10] and showed that the method A is the most accurate of the considered methods. When the pose is estimated using our method, the direct estimation of rotation is obtained using the first method given by (27), except when it is indicated that the Orthogonal Procrustes method has been used.

A. Results for pose tracking

In this paragraph, the ability of each method to track the pose of the camera with respect to a set of points for image sequences with abrupt motions is tested. A camera model with focal scaling factors $F_x = F_y = 800 \; pixels/m$ and principal point coordinates $u_x = v_x = 400 \; pixels$ has been used to compute the image points. For our method, the scalar gain λ has been set to 1.

The first sequence of 300 images is obtained using 9 non coplanar points defined in the object frame by (refer also to Fig. 5.a) :

$$\mathbf{X}_1=\begin{bmatrix} 0.2 & -0.2 & -0.2 & 0.2 & 0 & 0 & 0.1 & -0.13 & 0.4 \\ 0.2 & -0.2 & 0.2 & -0.2 & 0 & 0.15 & 0.01 & 0 & 0.4 \\ 1.01 & 1.02 & 0.96 & 1.03 & 1. & 1. & 1. & 1.2 & 1.3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{40}$$

White Gaussian noise with standard deviation equal to 0.5 has been added to the coordinates of each point in the image. Furthermore, the identity matrix has been used to initialize ${}^i\mathbf{M}_o$ for the first image of the sequence (the initial set of points is assumed to be in front of the camera close to the optical axis and at 1 $meter$ distance from the image plane). The computed pose for each image is used as initialization to determine the pose for the following one using each method. The evolution of the real pose parameters of the camera with respect to the object frame is shown in Fig. 6. Let us consider the pose error defined by :

$$\mathbf{T_e} = \begin{bmatrix} \mathbf{R_e} & \mathbf{t_e} \\ \mathbf{0_{1\times 3}} & 1 \end{bmatrix} = \mathbf{T_r^{-1}T_c}, \tag{41}$$

where $\mathbf{T_r}$ and $\mathbf{T_c}$ are respectively the real and the estimated poses. If the correct pose is obtained, $\mathbf{T_e}$ is equal to the identity matrix ($\| \mathbf{t_e} \| = 0$ and $\mathbf{R_e} = \mathbf{I_3}$). Let θ_e be the norm of the rotation vector $\theta_e\mathbf{u}$ corresponding to the rotation matrix $\mathbf{R_e}$ (recall that $\theta_e\mathbf{u}$ is linked to $\mathbf{R_e}$ by the Rodrigues' formula). The errors $\| \mathbf{t_e} \|$ and θ_e on the estimated poses using our method, method A and method L are shown respectively in Figs 7, 8 and 9. From these plots, it can be seen that the estimated values using the three methods are similar and very near to the real ones. Furthermore, the errors on the estimated pose obtained using the three methods are similar.

The second image sequence is obtained using less points (5 non-coplanar points) (refer to Fig. 5.b) :

$$\mathbf{X}_2 = \begin{bmatrix} 0.2 & -0.2 & -0.2 & 0.2 & 0 \\ 0.2 & -0.2 & 0.2 & -0.2 & 0.4 \\ 1.01 & 1.01 & 0.95 & 1.03 & 1. \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{42}$$

A stronger white gaussian noise with standard deviation equal to 2 has been added to the coordinates of each point. The results obtained using our method, method A and method L are shown respectively in Figs 11, 12 and 13. From Fig. 11, it can be seen that the estimated values of the pose using our method follow closely the real ones. On the other hand the method A diverged and was not able to estimate the pose (refer to Fig. 12). Finally, as it was mentioned in [20], method L is affected by local minima. Indeed from the plots, it can be noticed that the pose switched several times to local minima (refer to Fig. 13).

B. Convergence for random poses

In this paragraph, we compare the convergence rate for random poses using our method, method L and method A. The following setup has been used :

- an object composed of 8 coplanar points defined as follows has been considered (refer also to Fig. 5.c) :

$$\mathbf{X}_3=\begin{bmatrix} -0.4 & 0.4 & -0.4 & 0.4 & 0.42 & -0.09 & 0.32 & -0.32 \\ -0.4 & -0.4 & 0.4 & 0.4 & -0.28 & 0.32 & 0 & 0 \\ 1. & 1 & 1. & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{43}$$

- random poses have been generated as follow :
  - 1000 random rotational motions are firstly applied to the point coordinates defined in the object frame. The norm of the rotation around the x-axis and the y-axis range from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, while the rotation angle around the optical axis ranges from 0 to 2π.
  - for each generated rotation, a translational motion with respect to the optical axis that ranges from 1 $meter$ to 4 $meters$ is applied to the point coordinates defined in the object frame. Furthermore, the translational motion with respect to the x-axis and the y-axis are chosen such that the points coordinates belongs to the image limits $[1\ 800;\ 1\ 800]\, pixels$.

The error on the pose is calculated using $\| \mathbf{t_e} \|$ and θ_e computed from (41). Furthermore, for all methods, the identity matrix is used as the initial value of the pose matrix. Figures 14.a and 14.b give the distribution of $\| \mathbf{t_e} \|$ and θ_e using the three different methods and using perfect data (no noise on the point coordinates in the image). In other words, for each value of $\| \mathbf{t_e} \|$ and θ_e, the plot gives the percentage of the errors smaller or equal to these values. From these figures, it can be seen that our method achieves a convergence rate around 90%, while method L and A achieve convergence rates around 70% and 50% respectively. The case of non convergence to the global minimum using our method and method L are due to convergence to local minima. Conversely, in the case where the method A is used, the non convergences to the global minimum are due to both divergence and convergence to local minima.

Now, we test the convergence rate of the three methods using the same setup, but with 1 pixel gaussian noise on the point coordinates in the image. In this experiment, the method of rotation estimation by solving the Orthogonal

Procrustes Problem (OPP) is also tested. The results obtained using each method are given on Fig. 15. From this figure, it can be noticed that the accuracy of all the pose estimation methods decreased. However, our iterative method gives more accurate estimation of the poses. From Figure 15.b, it can be noticed that the rotation estimation by solving the OPP gives slightly better results than the first method to estimate the rotation. On the other hand, as mentioned above, the direct method recovering the rotation by building an orthonormal basis from a set of projected points has a dual use since it can also be used in image-based visual servoing [23], [5]. This does not seem possible using the formulation of rotation as an OPP.

As it has been shown above, the pose estimation can be performed as an iterative minimization without constraints for only three parameters that are the translation parameters t_x, t_y and t_z. This limits the space that has to be searched for to find the global optimum. Since the method we propose allows a high convergence rate, this makes possible preinitializing the iterative algorithm at random starting points. The low dimensionality of the space permits also to spot the local minima. Let us consider the following case where the object point coordinates are defined by (44) and the translational motion to be computed is defined by the vector $[0.7\ 0.4\ 2]m$. Figure 16 shows the cost function $\parallel \mathbf{s}_t^* - \mathbf{s}_t \parallel$ as color level for 5 values of t_z and t_x and t_y ranging from $-2m$ to $+2m$. From this Figure, we can spot the positions of the local and global minima with respect to each other (the position of the minimal value of the cost function for each value t_z is marked by a cross in the images). We also note from Figures 16.d and 16.e that no local minimums exist for $t_z > 2m$.

$$\mathbf{X}_4 = \begin{bmatrix} -0.4 & 0.4 & -0.4 & 0.4 & 0.5 \\ -0.4 & -0.4 & 0.4 & 0.4 & 0.6 \\ 1. & 1 & 1. & 1 & 1 \end{bmatrix} \qquad (44)$$

V. Conclusion and Future Work

In this paper, we have proposed a new pose estimation method from a set of matched points based on an invariant to rotations. The proposed method has been validated and compared to two different non-linear methods. The results obtained show that the proposed method achieves better tracking of the pose for image sequences but also a higher rate of convergence compared to the other methods considered. Future works will be devoted to extend this method to model-free pose estimation and also to camera calibration from a set of points.

L

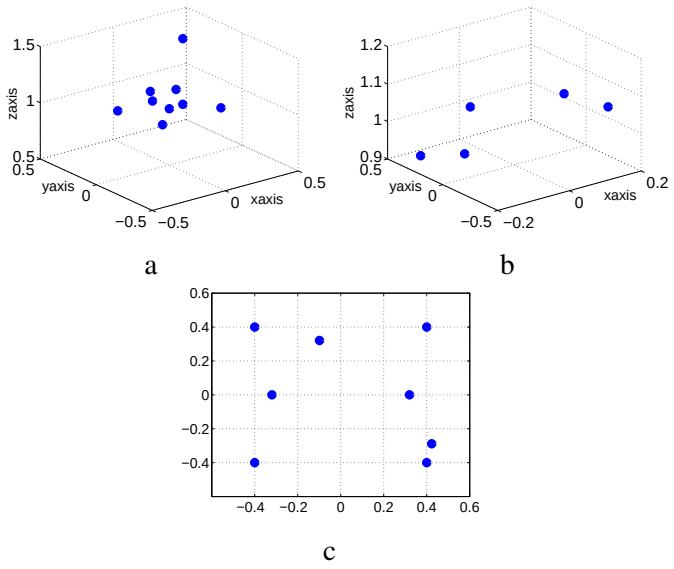


FIGURE 5. Object used for validations : a) object corresponding to $\mathbf{X}_1$, b) object corresponding to $\mathbf{X}_2$, c) planar object corresponding to $\mathbf{X}_3$.

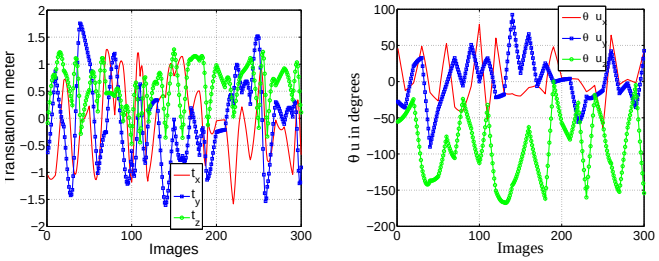


FIGURE 6. Real values of the pose for the image sequence 1 versus image number : left) translation vector entries in meter, right) rotation vector entries in degrees.

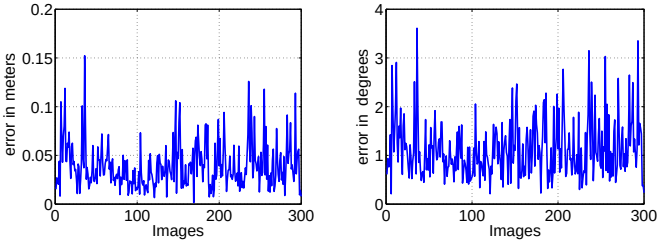


FIGURE 7. Error on the estimated pose parameters using our method for the image sequence 1 versus image number : left) $\parallel \mathbf{t_e} \parallel$, right) θ_e.

Appendix A : Proof of the Lemma 1

If only a rotational motion is considered between the two camera poses we have :

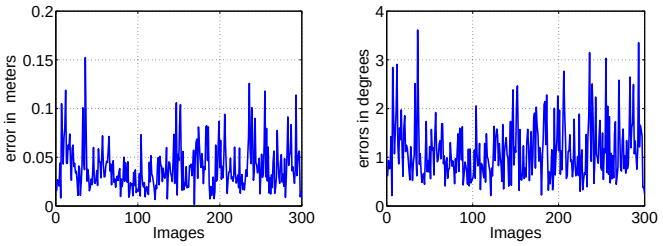


FIGURE 8. Error on the estimated pose parameters using method A for the image sequence 1 versus image number : left) $\| \mathbf{t_e} \|$, right) θ_e.

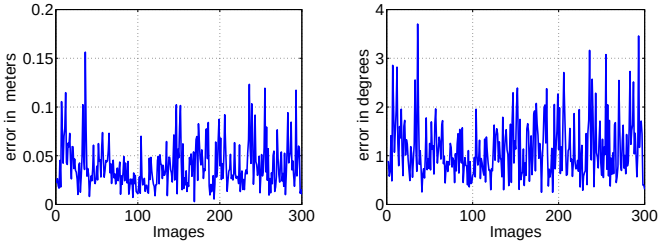


FIGURE 9. Error on the estimated pose parameters using method L for the image sequence 1 versus image number : left) $\| \mathbf{t_e} \|$, right) θ_e.

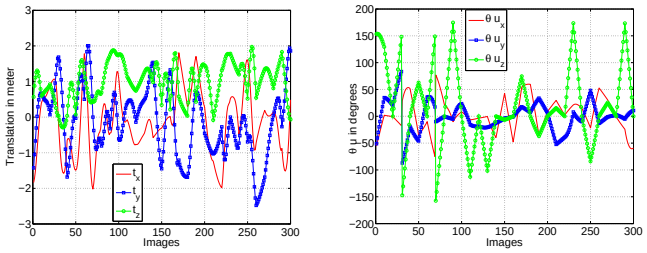


FIGURE 10. Real values of the pose for the image sequence 2 versus image number : left) translation vector entries in meter, right) rotation vector entries in degrees.

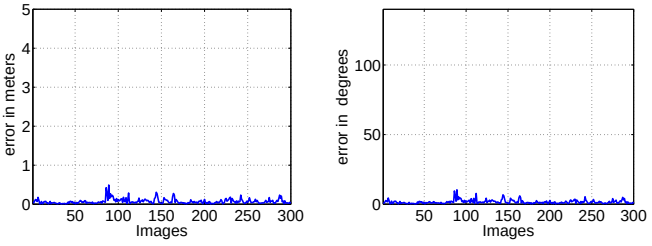


FIGURE 11. Error on the estimated pose parameters using our method for the image sequence 1 versus image number : left) $\| \mathbf{t_e} \|$, right) θ_e.

$$\mathbf{P}^*_{v_1} = \sum_{i=1}^{N} a_{1i}\mathbf{P}^*_{s_i} = \sum_{i=1}^{N} a_{1i}\, {}^{c*}\mathbf{R}_c \mathbf{P}_{s_i} = {}^{c*}\mathbf{R}_c\, \mathbf{P}_{v_1} \quad (45)$$

$$\mathbf{P}^*_{v_2} = \sum_{i=1}^{N} a_{2i}\mathbf{P}^*_{s_i} = \sum_{i=1}^{N} a_{2i}\, {}^{c*}\mathbf{R}_c \mathbf{P}_{s_i} = {}^{c*}\mathbf{R}_c\, \mathbf{P}_{v_2} \quad (46)$$

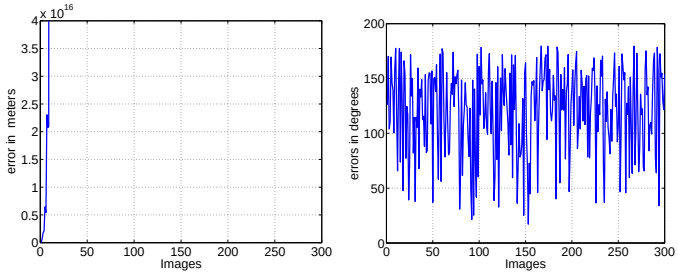


FIGURE 12. Error on the estimated pose parameters using method A for the image sequence 1versus image number : left) $\| \mathbf{t_e} \|$, right) θ_e.

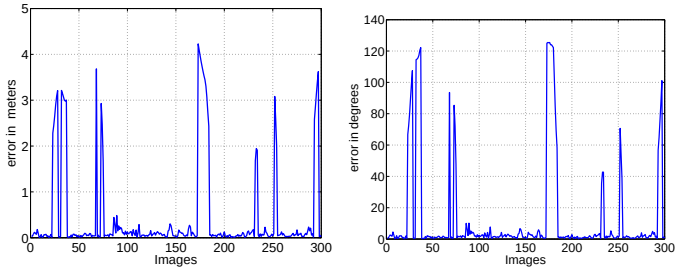
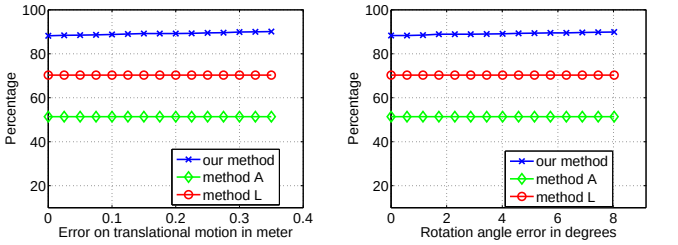


FIGURE 13. Error on the estimated pose parameters using method L for the image sequence 1 versus image number : left) $\| \mathbf{t_e} \|$, right) θ_e.



a b

FIGURE 14. Percentage of convergence with perfect data : a) $\| \mathbf{t_e} \|$, b) θ_e

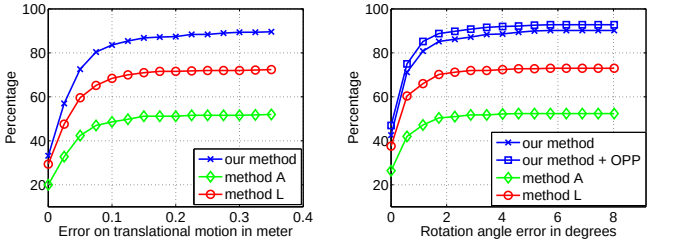


FIGURE 15. Percentage of convergence with 1 pixel gaussian noise on image point coordinates : left) $\| \mathbf{t_e} \|$, right) θ_e

By combining these equations with (24), we obtain :

$$\mathbf{v}^*_{n1} = \frac{{}^{c*}\mathbf{R}_c\, \mathbf{P}_{v_1}}{\sqrt{\mathbf{P}^\top_{v_1}\, {}^{c*}\mathbf{R}^\top_c\, {}^{c*}\mathbf{R}_c\, \mathbf{P}_{v_1}}} \quad (47)$$

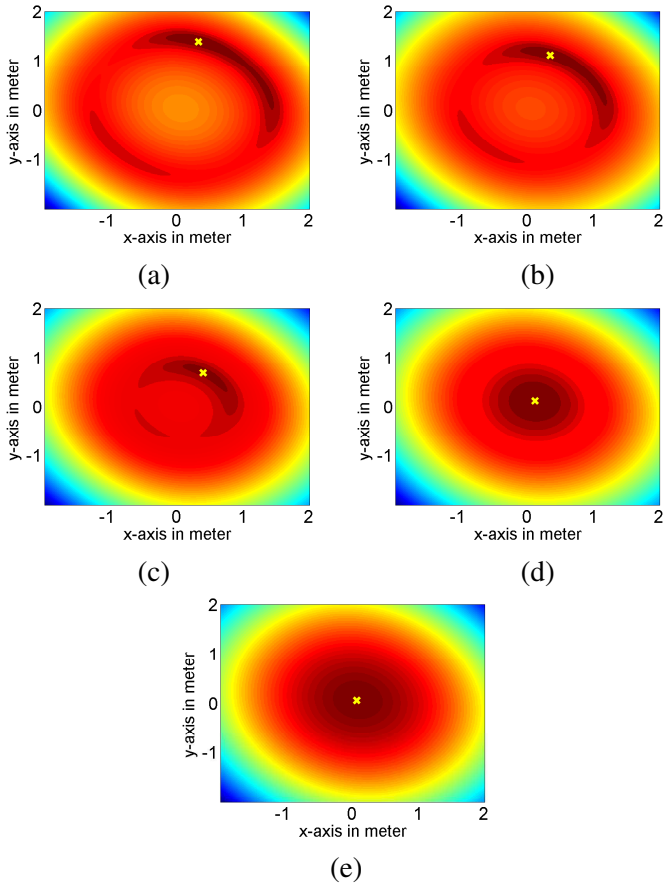


FIGURE 16. The cost function as a color level : (a) result for $t_z = 1.6m$, (b) result for $t_z = 1.8m$, (c) result for $t_z = 2m$, (d) result for $t_z = 2.2m$ (e) result for $t_z = 2.4m$

from which, we obtain :

$$\mathbf{v}^*_{n1} = {}^{c*}\mathbf{R}_c \, \mathbf{v}_{n1} \tag{48}$$

Identically, it is possible to prove that :

$$\mathbf{v}^*_{n2} = {}^{c*}\mathbf{R}_c \, \mathbf{v}_{n2} \tag{49}$$

Combining (48), (49) and (26) it yields :

$$\mathbf{v}^*_{n3} = {}^{c*}\mathbf{R}_c \, \mathbf{v}_{n3} \tag{50}$$

Therefore, by combining (48), (49) and (50), one obtains :

$${}^{c*}\mathbf{R}_{n*} = {}^{c*}\mathbf{R}_c \, {}^{c}\mathbf{R}_n \tag{51}$$

Finally, by multiplying both sides of the last equation by ${}^{c}\mathbf{R}_n^\top$, we obtain

$${}^{c*}\mathbf{R}_{n*} \, {}^{c}\mathbf{R}_n^\top = {}^{c*}\mathbf{R}_c \, {}^{c}\mathbf{R}_n \, {}^{c}\mathbf{R}_n^\top = {}^{c*}\mathbf{R}_c \tag{52}$$

ACKNOWLEDGMENT

We thank one of the anonymous reviewers for his suggestion about the Procrustes problem. This work was partly funded by grants from the French program " investissement d'avenir " managed by the National Research Agency (ANR), the European Commission (Auvergne FEDER funds) and the " Région Auvergne " in the framework of the LabEx IMobS3 (ANR-10-LABX-16-01) and by grants from PHC PESSOA 2011 (25116ND).

RÉFÉRENCES

[1] A. Ansar and K. Daniilidis. Linear Pose Estimation from Points or Lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5) :578–589, 2003.

[2] H. Araujo, R. L. Carceroni, and C. M. Brown. A Fully Projective Formulation to improve the Accuracy of LoweŠs Pose-Estimation Algorithm. *Computer Vision and Image Understanding*, 70 :227–238, 1998.

[3] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A Generic Fisheye Camera Model for Robotic Applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'07*, pages 1683–1688, San Diego, CA, USA, 2007.

[4] J. Courbon, Y. Mezouar, and P. Martinet. Evaluation of the Unified Model on the Sphere for Fisheye Cameras in Robotic Applications. *Advanced Robotics*, 6(8) :947–967, 2012.

[5] N. Cowan and D. E. Chang. Geometric Visual Servoing. *IEEE Transactions on Robotics*, 21(6) :1128 – 1138, Dec. 2005.

[6] D. Dementhon and L. Davis. Model-based Object Pose in 25 Lines of Code. *Int. Journal of Computer Vision*, 15(1-2) :123–141, June 1995.

[7] M. Dhome, M. Richetin, J.-T. Lapreste, and G. Rives. Determination of the Attitude of 3D Objects from a Single Perspective View. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12) :1265–1278, Dec 1989.

[8] P. Fiore. Efficient Linear Solution of Exterior Orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2) :140–148, Feb 2001.

[9] C. Geyer and K. Daniilidis. Mirrors in Motion : Epipolar Geometry and Motion Estimation. *International Journal on Computer Vision*, 45(3) :766–773, 2003.

[10] D. Grest, T. Petersen, and V. Krüger. *A Comparison of Iterative 2D-3D Pose Estimation Methods for Real-Time Applications*, volume 5575 of *Lecture Notes in Computer Sciences : Image Analysis*, pages 706–715. Springer, 2009.

[11] T. Hamel and R. Mahony. Visual Servoing of an Under-Actuated Dynamic Rigid Body System : an Image-Based Approach. *IEEE Transactions on Robotics and Automation*, 18(2) :187–198, April 2002.

[12] J. R. Hurley and R. B. Cattell. The procrustes program : Producing direct rotation to test a hypothesized factor structure. *Systems Research and Behavioral Science*, 7 :258–262, 1962.

[13] F. Janabi-Sharifi and M. Marey. A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing. *IEEE Transactions on Robotics*, 26(5) :939–947, October 2010.

[14] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP : An Accurate O(n) Solution to the PnP Problem. *International Journal on Computer Vision*, 12(2) :155–166, December 2009.

[15] D. G. Lowe. Fitting Parameterized Three-Dimensional Models to Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 :441–450, 1991.

[16] C.-P. Lu, G. Hager, and E. Mjolsness. Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6) :610–622, Jun 2000.

[17] C. Mei and P. Rives. Single View Point Omnidirectional Camera Calibration from Planar Grids. In *IEEE International Conference on Robotics and Automation*, pages 3945–3950, April 2007.

[18] R. Safaee-Rad, I. Tchoukanov, K. Smith, and B. Benhabib. Three-Dimensional Location Estimation of Circular Features for Machine Vision. *IEEE Transactions on Robotics and Automation*, 8(5) :624–640, Oct 1992.

[19] P. H. Schönemann. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 31(1) :1–10, March 1966.

[20] G. Schweighofer and A. Pinz. Robust Pose Estimation from a Planar Target. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 :2024–2030, 2006.

[21] A. Shademan and F. Janabi-Sharifi. Sensitivity Analysis of EKF and Iterated EKF Pose Estimation for Position-Based Visual Servoing. In *IEEE International Conference on Control Applications*, pages 755–760, Toronto, Canada, August 2005.

[22] O. Tahri and F. Chaumette. Complex Objects Pose Estimation Based on Image Moment Invariants. In *IEEE Int. Conf. on Robotics and Automation, ICRA'05*, pages 438–443, Barcelona, Spain, April 2005.

[23] R. Tatsambon Fomena, O. Tahri, and F. Chaumette. Distance-Based and Orientation-Based Visual Servoing from Three Points. *IEEE Transactions on Robotics*, 27(2) :256–267, April 2011.

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/) of the institute for Systems and Robotics, Coimbra, Portugal (http ://www.isr.uc.pt/). His research interests include mobile robotics, visual servoing and computer vision.

Helder Araujo is a Professor in the Department of Electrical and Computer Engineering of the University of Coimbra, Portugal and also an Invited Professor at Blaise Pascal University, Clermont-Ferrand, France. He is a researcher at the Institute for Systems and Robotics-Coimbra. His research interests include computer and robot vision, robot navigation and localization, and sensor modeling for robot navigation and localization.

Youcef Mezouar was born in Paris, France, in 1973. He received the Ph.D. degree in computer science from the University of Rennes 1, Rennes, France, in 2001. He was a Post-doctoral Associate in the Robotics Laboratory of the Computer Science Department, Columbia University, New York, NY. Since 2002, he has been with the Robotics and Vision Group, Institut Pascal Clermont-Ferrand, Aubiere, France. His current research interests include robotics, computer vision, vision-based control, and mobile robots

François Chaumette (SM'10) 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. He is currently the Head of the Lagadic Group (http ://www.irisa.fr/lagadic) at INRIA, centre Rennes-Bretagne Atlantique and IRISA, Rennes. 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. From 2001 to 2005, he has been an Associate Editor of the IEEE Transactions on Robotics. He is currently in the Editorial Board of the International Journal of Robotics Research.