2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

On the use of IMUs in the PnP Problem

Luigi D'Alfonso, Emanuele Garone, Pietro Muraca, Paolo Pugliese

***Abstract*— In this paper the problem of estimating the relative orientation and position between a camera and an object is investigated. It is assumed that both the camera and the object are provided with an Inertial Measurement Unit (IMU) capable of measuring their attitude with respect to the gravity and the earth magnetic vectors. Furthermore, the object is assumed to contain a feature of n points, the position of which is known in the object coordinate frame. An algorithm is proposed, which uses the image provided by the camera and the information provided by the IMUs, to solve the PnP problem, i.e., to estimate the relative pose of the object in the camera reference frame. Two special cases will be studied. The first is the case where all the attitude information given by the IMU is used. In the second case only the measurements provided by inclinometers are used, neglecting those coming from the magnetometers, because they are usually quite noisy. The effectiveness of the proposed algorithms is tested either by numerical simulations and by experimental tests with cameras.**

I. Introduction

The Perspective-n-Point (PnP) problem, also known as pose estimation, has been introduced for the first time in the 80's by Fischler and Bolles [1]. It is the problem of determining the relative position and orientation of an object with respect to a camera by exploiting the image provided by the camera and the knowledge of a feature of n points placed on the object (see Figure 1). This problem finds application in several fields, such as computer vision [2], computer animation [3], photogrammetry [4] and robotics [5], [6], [7].

Several solutions to the PnP problem have been proposed in the literature. From the theoretical viewpoint, it has been proved that the smallest number of points which yield to a finite number of solutions for this problem is $n = 3$ [1], since the P2P problem ($n = 2$), in its classical formulation, admits infinite solutions. Moreover, as proved in [8], the smallest number of feature points ensuring a unique solution to the PnP problem in all possible object configurations is $n = 4$, under the assumption they are coplanar and no more than two of them lie on a single line.

Even if using $n = 4$ correspondences between feature points and pixels is sufficient to obtain a unique solution for the PnP problem, a larger set of points, say $n > 4$, makes the solution more robust with respect to the measurement noise. For this reason, many works in the literature focus on finding algorithms to solve the generic PnP problem for $n > 4$. In [9], authors consider triplets of points among the n available correspondences, and for each of them they derive fourth degree polynomials in the unknowns of the problem. Then, they rearrange such polynomials in a matrix form and use the singular value decomposition to estimate the unknown values and solve the PnP problem. In [10], authors propose a non-iterative solution to the PnP problem, the computational complexity of which grows linearly with n. The main idea behind this method is to express the n feature points as a weighted sum of four virtual control points in the camera reference frame. The control points can be estimated in $O(n)$ time by expressing their coordinates as weighted sum of the eigenvectors of a 12×12 matrix and solving a small number of quadratic equations to pick the right weights. In [11], an iterative method based on the minimization of an error index in the 3D space, optimizing alternatively on the relative position and orientation unknowns, is proposed. This algorithm has a very fast convergence time, but it can get stuck in local minima, depending on its initial guess.

In the classical PnP problem the only available information is the one provided by the camera and the feature. However, in many cases, such as in robotics applications, additional sensors are available that might provide further useful information to enhance the pose estimation. For instance, mobile robots and cameras are often equipped with Inertial Measurement Units (IMUs) that, in the static configuration, are able to measure the gravity vector, in their own reference frame, through the accelerometers, and the magnetic field, by the magnetometers. Therefore, at the equilibrium, an IMU provides all the information on the rotation of the sensor itself with respect to North-East-Down (NED) reference frame. However, it should be remarked that the information on the magnetic field is quite unreliable as magnetometers are in general imprecise as their measurements are often affected by local magnetic fields.

The aim of this paper is to use data from the IMUs to 'help' the vision system to solve the PnP problem. In particular, we will consider a static configuration where, due to the high resolution of the IMUs' accelerometers, the proposed approach is expected not only to simplify the solution of the PnP problem, but also to yield a more accurate pose estimation.

To the best of our knowledge, only a few recent works using this approach have been presented in the literature. In [12], the authors use the roll, pitch and yaw angles provided by the IMU to compute the translation vector between the feature and a fixed camera with known position and orientation. In the same paper it is also shown that, if the whole object attitude is known, the P2P problem admits a unique solution whenever the two points on the image are distinct. However, it is worth mentioning that the results

E. Garone is with Faculty of Applied Science, Control and Systems Analysis Department, Université Libre de Bruxelles, 1050 Bruxelles, Belgium. E-mail: *egarone@ulb.ac.be.*

L. D'Alfonso, P. Muraca, P. Pugliese are with DIMES, Università della Calabria, Rende, E-mail: {*ldalfonso,muraca,pugliese*}*@dimes.unical.it.*

obtained in that paper, although interesting from the scientific viewpoint, are quite fragile w.r.t. pixel noise. Moreover, the information about the attitude is computed by the IMU using both the accelerometers and the compass, the latter of which, as already remarked, can be noisy and unreliable. In [13], authors assume to know the coordinates of two points in the absolute reference frame and, using the roll and pitch provided by an IMU mounted on the camera, they solve a P2P problem on reconstructing the absolute camera pose. The yaw angle provided by the IMU is discarded due to the low accuracy of the compass.

In the present paper we consider a more general scenario where no assumption on the absolute coordinates of either the camera or the object is made, and both the observed object and the camera are provided with IMUs. Several results obtained for this configuration will be presented, both in the case where the information of the magnetometer is used or only inclinometers are considered.

Early theoretical results along this line have been presented in [14] and [15]. In these works it has been shown that the P2P problem using only inclinometers always gives two solutions, except when the two points are seen as one by the camera. Moreover, it has been remarked that singular configurations and ambiguities could be avoided by using a feature of 3 non-collinear points.

In this paper, first we will propose a closed-form solution for the P2P Problem with IMUs (i.e., using both inclinometers and magnetometers) which considerably improves the robustness to pixel noise w.r.t. the algorithm proposed in [12]. Then we will extend the above solution to the general PnP Problem with IMUs.

Finally, the PnP problem using only inclinometers will be investigated. In particular, the PnP problem will be reformulated as an optimization problem where the objective function is a least mean square index. This choice allows to solve the PnP problem by solving a single variable unconstrained optimization problem. The effectiveness of the proposed PnP problem solution will be tested by means of both numerical simulations and experimental tests.

The paper is organized as follows: in Section II we state the problem and define our framework; in Section III the solution to the PnP Problem with the rotation matrix provided by IMUs is proposed; in Section IV the solution to the PnP problem using only accelerometers is described and analyzed; in Section V the experimental results are shown and in Section VI we draw our conclusions.

II. Problem Statement

Assume to have a camera and an object in the field of view of the camera. Two reference frames are defined: the camera reference frame O_{xyz}, the origin of which is in the camera focus, and the object reference frame O'_{uvw}. It is assumed that the object is provided with a feature composed of n points, the coordinates of which are known a priori in the object reference frame. Both the object and the camera are equipped with an IMU capable of measuring

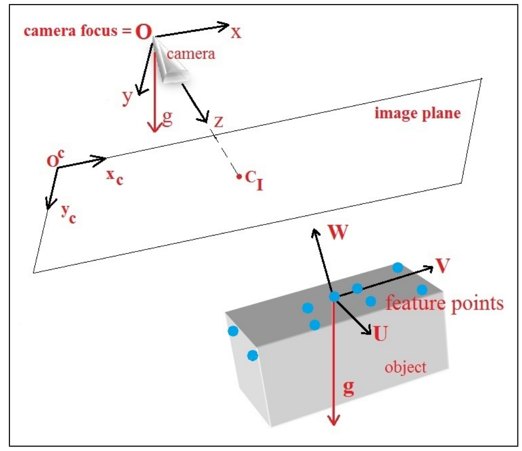

Fig. 1. Camera, image and object reference frames

the gravity unit vector[1] and the earth magnetic field unit vector: $\hat{g}_o = [g_u, g_v, g_w]$, $\hat{m}_o = [m_u, m_v, m_w]$ in the object reference frame, and $\hat{g}_c = [g_x, g_y, g_z]$, $\hat{m}_c = [m_x, m_y, m_z]$ in the camera reference frame, respectively. It is assumed that the distance between the camera and the object is negligible w.r.t. the earth radius, hence $\hat{g}_c, \hat{m}_c$ and $\hat{g}_o, \hat{m}_o$ represent the same couple of vectors in two different coordinate frames. Furthermore, the camera is assumed to have no distortion and to be characterized by the focal length f and the distance per pixel dpx. The image reference frame is denoted by $O^c_{x_c y_c}$, the image plane is $z = f$ and the image center is $C_I = (x_{C_I}, y_{C_I})$. The overall scenario is shown in Figure 1.

The goal of this paper is to solve the PnP problem using information provided both by the IMUs and the camera. The aim is to obtain the transformation matrix from the object reference frame to the camera reference frame

$$R_t = \left[\begin{array}{ccc|c} & R & & t \\ 0 & 0 & 0 & 1 \end{array}\right],$$

where R is the relative rotation matrix and $t = [t_x, t_y, t_z]^{\mathrm{T}}$ is the translation vector.

III. PnP Problem using IMUs

Using the information provided by the IMUs, the whole rotation matrix from the object reference frame to the camera reference frame can be estimated. In this context, solving the PnP problem translates into determining the translation vector t between the two reference frames. In the following, a new solution to the P2P problem with IMUs will be presented. Afterwards, this solution will be extended to the more general PnP problem with IMUs.

A. P2P Problem with known Rotation Matrix

The P2P problem is the problem of estimating the relative pose of the object in the camera reference frame when the object contains a feature of two distinct points $A = (A_u, A_v, A_w)$ and $B = (B_u, B_v, B_w)$, the coordinates

[1]The IMU also gives the magnitude of the gravity and earth magnetic field vectors, but in this work only the vectors directions are used.

of which are known in the object reference frame. The information provided by the camera can be modeled using the pinhole model [16] as follows. Let $K = (K_u, K_v, K_w)$ be a point in the object reference frame. Using the transformation matrix R_t, the coordinates of K in the camera reference frame can be written as

$$P_K = [K_x, K_y, K_z]^T = RK + t, \tag{1}$$

the coordinates of pixel P_K^* related to P_K in the image being

$$x_K = \tilde{f}\,\frac{K_x}{K_z} + x_{C_I}, \quad y_K = \tilde{f}\,\frac{K_y}{K_z} + y_{C_I}, \tag{2}$$

where $\tilde{f} = f/dpx$ is the camera focal length in pixels. Using the above equations it is possible to compute the pixels $P_A^* = (x_A, y_A)$ and $P_B^* = (x_B, y_B)$, where points A and B are projected in the image plane. On defining $\tilde{P}_A = P_A^*/\tilde{f}$, $\tilde{P}_B = P_B^*/\tilde{f}$, from equations (2) and (1) we obtain

$$\begin{aligned} \tilde{P}_A &= (\tilde{x}_A, \tilde{y}_A) = \left(\frac{r_1 A + t_x}{r_3 A + t_z}, \frac{r_2 A + t_y}{r_3 A + t_z} \right), \\ \tilde{P}_B &= (\tilde{x}_B, \tilde{y}_B) = \left(\frac{r_1 B + t_x}{r_3 B + t_z}, \frac{r_2 B + t_y}{r_3 B + t_z} \right), \end{aligned} \tag{3}$$

where r_i is the i-th row of matrix R.

Since all the information from the IMUs is used, matrix R is known, thus solving the P2P problem simplifies to estimate the translation vector t. A possible way to find t is to use the results presented in [12]. There, the authors solve equations (3) with respect to t_x, t_y, t_z, obtaining the solution

$$\begin{aligned} t_x &= A_z \tilde{x}_A - r_1 A, \\ t_y &= A_z \tilde{y}_A - r_2 A, \\ t_z &= A_z - r_3 A, \end{aligned} \tag{4}$$

where

$$A_z = \begin{cases} \dfrac{(r_1 - \tilde{x}_B r_3)(A - B)}{\tilde{x}_A - \tilde{x}_B} & \text{if } \tilde{x}_B \neq \tilde{x}_A, \\ \dfrac{(r_2 - \tilde{y}_B r_3)(A - B)}{\tilde{y}_A - \tilde{y}_B} & \text{otherwise.} \end{cases} \tag{5}$$

The above solution is very sensitive to pixel noise. To mitigate the problem, we compute the translation vector $\hat{t}$ which minimizes the following index

$$\mathbb{E} = ||\hat{P}_A(\hat{t}) - \tilde{P}_A||^2 + ||\hat{P}_B(\hat{t}) - \tilde{P}_B||^2, \tag{6}$$

where $\hat{P}_A(\hat{t})$ and $\hat{P}_B(\hat{t})$ are the pixels in which points A and B are projected using the rotation matrix R provided by the IMUs, the translation vector $\hat{t}$ and equations (3). The above index represents the reprojection error due to using R and $\hat{t}$ ([12], [13], [10]). It is important to highlight that the resulting optimization problem must be solved through numerical methods and may require a high computational effort. To overcome this difficulty, an alternative least mean square index is proposed. The starting point is that equations (3) can be rewritten as the following set of linear equations

$$\begin{aligned} -t_x + t_z \tilde{x}_A &= r_1 A - r_3 A \tilde{x}_A, \\ -t_y + t_z \tilde{y}_A &= r_2 A - r_3 A \tilde{y}_A, \\ -t_x + t_z \tilde{x}_B &= r_1 B - r_3 B \tilde{x}_B, \\ -t_y + t_z \tilde{y}_B &= r_2 B - r_3 B \tilde{y}_B, \end{aligned} \tag{7}$$

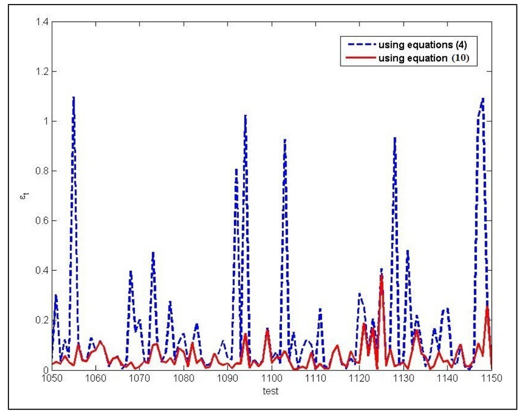

Fig. 2. Index ε_t in some of the performed tests

which can be written in matrix form as follows

$$Q\,t = s, \tag{8}$$

where

$$Q = \begin{bmatrix} -1 & 0 & \tilde{x}_A \\ 0 & -1 & \tilde{y}_A \\ -1 & 0 & \tilde{x}_B \\ 0 & -1 & \tilde{y}_B \end{bmatrix}; \quad s = \begin{bmatrix} r_1 A - r_3 A \tilde{x}_A \\ r_2 A - r_3 A \tilde{y}_A \\ r_1 B - r_3 B \tilde{x}_B \\ r_2 B - r_3 B \tilde{y}_B \end{bmatrix}. \tag{9}$$

An index alternative to (6) is defined as

$$\mathbb{E}_2 = ||Q\,t - s||^2, \tag{10}$$

which is the least mean square index over equations (7). The main advantage of using index (10) rather than (6) is that, as well known, the optimal solution of this LMS problem can be computed by the pseudo-inverse matrix of Q:

$$\hat{t} = Q^{\dagger} s. \tag{11}$$

Clearly, this solution is more robust with respect to pixel noise than the one provided by (4).

To verify the robustness of the proposed solution with respect to pixel noise, a set of 10 000 numerical tests has been performed, using a feature of $n = 2$ points placed in $A = [0, 0, 0]^T$ m and $B = [0.1, 0.1, 0]^T$ m in the object reference frame, assuming a zero mean Gaussian noise with standard deviation $\sigma = 5$ pixels on the pixels provided by the camera. For each test, the transformation matrix has been chosen randomly. The estimation of vector t by (11) has been contrasted with the one proposed in [12] by the relative error $\varepsilon_t = ||t - \hat{t}||/||t||$. Figure 2 shows the results of some of the performed tests. As expected, using the pseudo-inverse, the estimation error ε_t is generically smaller than the one obtained using equations (4) of [12], and provides an average value of 0.0691 (7% error) against an average value of 0.5029 (50% error) for the estimation method of [12].

Remark 1 Interestingly enough, index $\mathbb{E}$ and index $\mathbb{E}_2$ are related as $\mathbb{E} = ||M||^2\,\mathbb{E}_2$, where

$$M = \mathrm{diag}\left[\tfrac{1}{r_3 A + t_z}, \tfrac{1}{r_3 A + t_z}, \tfrac{1}{r_3 B + t_z}, \tfrac{1}{r_3 B + t_z} \right];$$

for this reason, and under appropriate assumptions (*e.g.*, $t_z \gg r_3(A - B)$), the optimal value for $\mathbb{E}_2$ can be taken as a good approximation of the optimal value for $\mathbb{E}$.

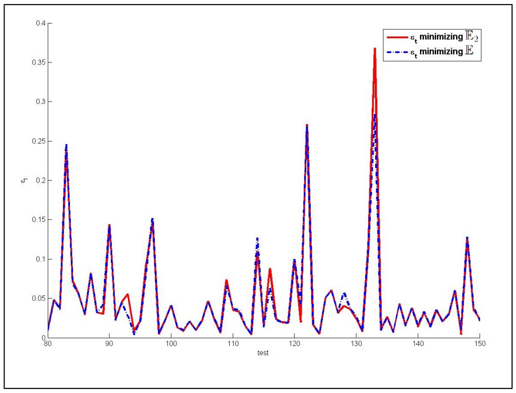

Fig. 3. Index ε_t minimizing $\mathbb{E}$ and minimizing $\mathbb{E}_2$

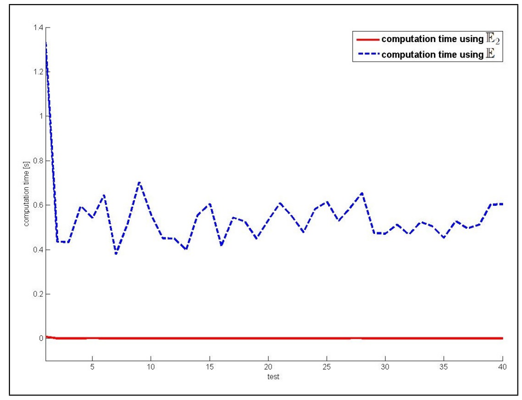

Fig. 4. Computation times required to minimize w.r.t. $\mathbb{E}$ or w.r.t. $\mathbb{E}_2$

Remark 2 As the index $\mathbb{E}$ is the most used in the literature when it comes to pose estimation, it is worth to compare it with $\mathbb{E}_2$. Figure 3 shows the ε_t index for some of the performed tests, computed using the translation vector obtained by equation (11) and the translation vector resulting from the index $\mathbb{E}$ minimization. Results show no evidence of any real estimation advantages in using $\mathbb{E}$ and $\mathbb{E}_2$. Conversely, as shown in Figure 4, the use of $\mathbb{E}_2$ clearly outperforms the use of $\mathbb{E}$ in terms of computational time. The averaged computation time[2] required to solve the P2P problem over the performed $10\,000$ tests is $0.0021s$ and $0.5563s$, minimizing w.r.t. $\mathbb{E}_2$ and $\mathbb{E}$ respectively.

B. PnP Problem with known rotation matrix

In this section the results obtained for the P2P problem will be extended to the PnP problem in the general case $n \geq 3$. To avoid situations where all the points of the feature P_i are seen by the camera as a single pixel it will be assumed that at least three points of the feature are not collinear.

As in the P2P case, if the rotation matrix R between the object and the camera reference frames is provided by IMUs, the PnP problem can be reformulated as a set of linear equations and LMS solution can be found using the pseudo-inverse method. More formally, given a set of n correspondences between feature points and pixels, the Q matrix and the s vector will contain $2n$ rows, two rows for each correspondence. Given a generic pixel $\tilde{P}_i = (\tilde{x}_i, \tilde{y}_i)$ and the related point $P_i = (P_{u,i}, P_{v,i}, P_{w,i})$ in the object reference frame, the Q matrix and the s vector will contain the following two rows, respectively:

$$\begin{aligned} Q_{i,1} &= [-1\ ,\ 0\ ,\ \tilde{x}_i] \quad s_{i,1} = r_1 P_i - r_3 P_i \tilde{x}_i, \\ Q_{i,2} &= [0\ ,\ -1\ ,\ \tilde{y}_i] \quad s_{i,2} = r_2 P_i - r_3 P_i \tilde{y}_i. \end{aligned}$$

Thus the matrix Q and the vector s will be:

$$Q = \begin{bmatrix} Q_{1,1} \\ Q_{1,2} \\ \vdots \\ Q_{n,1} \\ Q_{n,2} \end{bmatrix}, \quad s = \begin{bmatrix} s_{1,1} \\ s_{1,2} \\ \vdots \\ s_{n,1} \\ s_{n,2} \end{bmatrix}. \tag{12}$$

[2]The computation time refers to using Matlab R2012 running on an Intel Core i7 CPU Q720 processor.

Remark 3 Note that the dimension of matrix Q only grows linearly with n. Moreover, in the common case where the feature points are known a priori, the pseudo-inverse matrix $Q^\dagger$ can be pre-computed offline, thus translating the PnP problem in a mere matrix-vector product.

IV. PnP PROBLEM USING ACCELEROMETERS ONLY

In the previous Section, the rotation matrix provided by the IMUs is computed using both the magnetometers and the accelerometers. However, due to the low reliability of the compass, the information about the magnetic field can be noisy and yield to unreliable attitude values. In these cases, a possible choice is to neglect the measurements provided by the magnetometer and only use the inclinometers. To this end, the first step will be to parametrize the transformation matrix R_t with respect to the neglected data. Then, using this new transformation matrix, a method to tackle the PnP problem will be detailed.

Following the lines shown in [14], the rotation matrix R can be parametrized w.r.t. the angle α representing the information usually provided by the magnetometers. The resulting parametrized transformation matrix will be:

$$R_t(\alpha, t) = \left[\begin{array}{ccc|c} & R(\alpha) & & t \\ 0 & 0 & 0 & 1 \end{array}\right]. \tag{13}$$

Solving the PnP problem only using information provided by accelerometers translates in the problem of finding the angle α of the parametrized rotation matrix $R(\alpha)$ and the translation vector t between the camera and the object.

Given the set of feature points $\{P_i\}_{1=1}^n$ in the object reference frame and the related set of pixels $\{\tilde{P}_i\}_{i=1}^n$, the following index is defined

$$J(\alpha, t_x, t_y, t_z) = \sum_{i=1}^{n} ||\hat{P}_i(\alpha, [t_x, t_y, t_z]^T) - \tilde{P}_i||^2, \tag{14}$$

where $\hat{P}_i(\alpha, [t_x, t_y, t_z]^2)$ is the pixel in which the feature point P_i is mapped using the transformation matrix $R_t(\alpha, [t_x, t_y, t_z]^T)$.

The above index represents the total quadratic reprojection error due to the use of the transformation matrix $R_t(\alpha, [t_x, t_y, t_z]^T)$.

The following optimization problem can be formulated:

$$\{\alpha^*, t_x^*, t_y^*, t_z^*\} = \arg \min_{\alpha, t_x, t_y, t_z} J(\alpha, t_x, t_y, t_z); \tag{15}$$

once the optimal solution $\alpha^*, t_x^*, t_y^*, t_z^*$ has been found, the transformation matrix $R_t(\alpha^*, [t_x^*, t_y^*, t_z^*]^T)$ will be the solution to the PnP problem.

As explained in Section III, finding an optimal solution for the reprojection index may be difficult and computationally onerous. Again, to overcome this problem the use of LMS as an alternative cost function is proposed

$$J_2(\alpha, t_x, t_y, t_z) = ||Q(\alpha)t - s(\alpha)||^2, \tag{16}$$

where $Q(\alpha)$ and $s(\alpha)$ are the Q matrix and the s vector obtained in (12) when the rotation matrix is $R = R(\alpha)$. Using this function, the optimization problem associated to the solution of the PnP problem becomes

$$\{\alpha^*, t_x^*, t_y^*, t_z^*\} = \arg \min_{\alpha, t_x, t_y, t_z} J_2(\alpha, t_x, t_y, t_z). \tag{17}$$

Interestingly enough, this four variables optimization problem can be reformulated without loss of generality as an equivalent single variable optimization problem. In fact, if the angle α is fixed, the translation vector $\hat{t}$ minimizing the cost function admits the closed-form solution (11). Hence, $\hat{t}$ can be rewritten as a function of α, and Problem (17) can be equivalently reformulated as

$$\alpha^* = \arg\min_{\alpha} J_2(\alpha, \hat{t}_x(\alpha), \hat{t}_y(\alpha), \hat{t}_z(\alpha)), \tag{18}$$

where $\left[\hat{t}_x(\alpha), \hat{t}_y(\alpha), \hat{t}_z(\alpha)\right]^T = Q(\alpha)^\dagger s(\alpha)$. Clearly, the optimal solution of (18) will be the optimal solution for Problem (17) too. The main advantage of this reformulation is that the optimization problem (18) is a single variable unconstrained optimization problem whose solution is computationally less cumbersome than the one of (17).

Several efficient unconstrained optimization algorithms exist to solve the optimization problem (18), *e.g.*, the Nelder-Mead method [21]. Most of these algorithms are iterative and require an initial guess; thus, to solve (18), an initialization value α_0 is required. A possible estimate consists of solving a P3P problem on three arbitrary chosen feature points using the P3P problem solution proposed in [14].

V. Experimental Results

To assess the performance of the proposed algorithm, several numerical simulations and real experiments have been performed. Three algorithms have been compared: the method (18) proposed in this paper, implemented through the Nelder-Mead algorithm (using the Matlab function *fminsearch*) and hereafter denoted as α-PnP; the efficient PnP algorithm proposed in [10], denoted as e-PnP, in the implementation provided by in [20]; the PnP with IMUs method proposed in Section III-B, denoted as IMU-PnP and based on the rotation matrix provided by IMUs and on the translation vector computed using (11). To compare the results obtained by the three algorithms the relative error index $\varepsilon = ||R_t - \hat{R}_t||/||R_t||$ has been used, where $\hat{R}_t$ is the estimated transformation matrix obtained by one of the estimation algorithms and R_t is the real one.

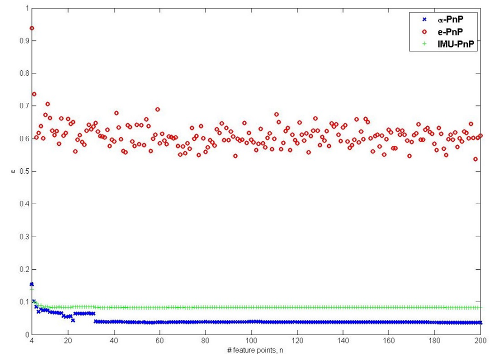

Fig. 5. Averaged ε index over 300 configurations with $n = 4, 5, \dots, 200$ feature points

A. Numerical Simulations

To evaluate the performance of the proposed solutions to the PnP problem, a set of simulations have been performed. In all simulations, different features with points randomly generated in the box $B = [-0.2, 0.2]^3$ have been used. To ensure the existence of at least three non collinear feature points, the first three points of each feature are in all cases $P_1 = [0, 0, 0]$, $P_2 = [0.1, 0.1, 0]$ and $P_3 = [0.1, 0, 0]$. For each test, the relative rotation and translation between the camera and the object reference frames have been randomly generated in the box $B_0 = [-0.5, 0.5]^2 \times [0.5, 2.5]$.

In all tests, a camera having a focal length $\tilde{f} = 800$ pixels, a resolution of 640×480 pixels and center $C_I = (320, 240)$ has been simulated. The data acquired by the IMUs and the image provided by the camera are affected by zero mean Gaussian noise with standard deviation $\sigma_g = 0.01\, I_3$ for $\hat{g}_c$ and $\hat{g}_o$, $\sigma_{pixel} = 5$ pixels in the image, $\sigma_{mag} = 8°$ on the measurements provided by the magnetometer. The performance of the algorithms have been tested with different values of the number n of feature points, starting from $n = 4$ up to $n = 200$. Each set of feature points has been tested in 300 different configurations. Figure 5 shows the averaged value of the ε index as n changes.

Note that the α-PnP algorithm performs better than the e-PnP and the IMU-PnP algorithms, always giving a smaller value of the ε index. Moreover, after a certain number of points (about 40), adding further points to the feature does not increase significantly the quality of the solution.

B. Real Experiments

To experimentally test the proposed PnP solutions, the following experimental setting has been used:

- A Logitech C310 HD webcam with resolution 1280×960 [17]. The camera intrinsic parameters have been estimated using the Camera Calibration Toolbox [19].
- Two ArduIMU V3 Inertial Measurement Units [18].
- The four squares feature shown in Figure 6. Each corner of the squares can be used as a feature point.

A set of experiments has been performed using the following procedure: as a first step, the camera and the object have been placed in an unknown configuration, with the object

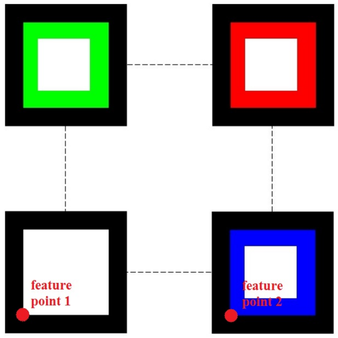

Fig. 6. The four squares feature

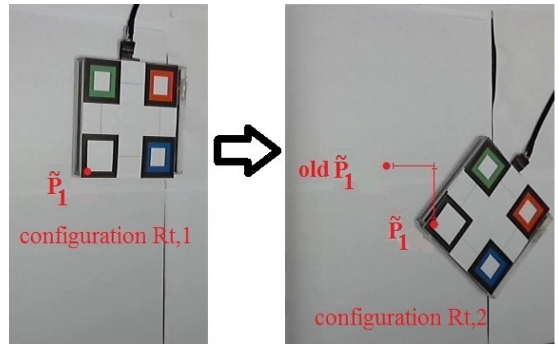

Fig. 7. Pictures taken by the camera in configuration $R_{t,1}$ (l) and $R_{t,2}$ (r)

in the field of view of the camera, and an estimation $\hat{R}_{t,1}$ of the actual transformation matrix $R_{t,1}$ has been computed using the α-PnP, the IMU-PnP and the e-PnP algorithms. Then the object has been rotated and translated by a known transformation matrix $R_{t,2}$. As a third step, an estimate of $R_{t,2}$, namely $\hat{R}_{t,2}$, has been obtained. The displacement matrix between the two configurations has been computed as $\hat{R}^2_{t,1} = (\hat{R}_{t,1})^{-1}\hat{R}_{t,2}$. Finally the estimation $\hat{R}^2_{t,1}$ has been compared with $R^2_{t,1}$. Figure 7 shows two pictures taken during the experiments.

The performance of the three algorithms have been contrasted through the proposed index ε using $n = 4, \ldots, 16$ feature points. A set of 42 experiments has been performed changing the relative rotation and translation between the camera and the object. The ε index has been computed over the obtained $R^2_{t,1}$ and $\hat{R}^2_{t,1}$. The resulting averaged ε index is $\varepsilon = 29\%$ for the e-PnP algorithm, $\varepsilon = 34\%$ for the IMU-PnP algorithm and $\varepsilon = 24\%$ for the α-PnP algorithm. Again the α-PnP algorithm presented in this paper performs in the average better than the e-PnP and the IMU-PnP algorithms.

VI. Conclusions

In this paper the problem of estimating the relative orientation and position between a camera and an object has been considered. It is assumed that both the camera and the object are equipped with an IMU and that the object contains a feature of n points, the position of which in the object reference frame is known a priori. Two new algorithms that solve the PnP problem using the image provided by the camera and the information measured by the IMUs have been presented. The first covers the case where all the information provided by the IMU is used. The second algorithm covers the case where only the measurements provided by inclinometers are used and the ones from the magnetometers are discarded. It has been shown that if the entire information provided by IMUs is used, then the resulting PnP problem solution is more affected by noise than the solution obtained using only cameras and inclinometers. Numerical simulations and experimental tests have shown the effectiveness of the proposed solution for the PnP problem.

References

[1] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM*, 24(6):381–395, 1981

[2] R. Horaud, B. Conio, O. Leboulleux, An analytic solution for the perspective 4-point problem, *Computer Vision, Graphics and Image Processing*, 47:33–44, 1989

[3] C. Su, Y. Xu, H. Li, S. Liu, Application of Wu's method in computer animation, *Proc. Fifth Int. Conf. on CAD/CG*, Shenzhen, China, December 2-5, 1997, pp. 211–215

[4] J.S.C. Yuan, A general photogrammetric method for determining object position and orientation, *IEEE Trans. Robot. Autom.*, 5(2):129–142, 1989

[5] G. Chen, D. Xu, P. Yang, High precision pose measurement for humanoid robot based on PnP and OI algorithms, *Proc. IEEE Int. Conf. on Robotics and Biomimetics*, Tianjin, China, December 14-18, 2010, pp. 620–624

[6] Q. Wang, X Zhang, D, Xu, Human behavior imitation for a robot to play table tennis, *Proc. 24th Chinese Control and Decision Conf.*, Taiyuan, China, May 23-25, 2012, pp. 1482–1487

[7] Q. Wang, X Zhang, D. Xu, On pose recovery for generalized visual sensors, *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(7):848–861, 2004

[8] M.A. Abidi, T. Chandra, A new efficient and direct solution for pose estimation using quadrangular targets; algorithm and evaluation, *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(5):534–538, 1995

[9] L. Quan, Z. Lan, Linear N-point camera pose determination, *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):774–780, 1999

[10] V. Lepetit, F. Moreno-Noguer, P. Fua, EPnP: An accurate $O(n)$ solution to the PnP problem, *Int. J. Comput. Vision*, 81(2):155–166, 2009

[11] C.P. Lu, G.D. Hager, E. Mjolsness, Fast and globally convergent pose estimation from video images, *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(6):610–622, 2000

[12] L. Merckel, T. Nishida, Evaluation of a method to solve the perspective-two-point problem using a three-axis orientation sensor, *Proc. 8th IEEE Int. Conf. on Computer and Information Technology*, Sydney, Australia, July 8-11, 2008, pp. 862–867

[13] Z. Kukelova, M. Bujnak, T. Pajdla, Closed-form solutions to the minimal absolute pose problems with known vertical direction, *Proc. 10th Asian Conf. on Computer Vision*, Queenstown, New Zeeland, November 8-12, 2010, pp. 216–229

[14] L. D'Alfonso, E. Garone, P. Muraca, P. Pugliese, P3P and P2P problems with known camera and object vertical directions, *Proc. 21st IEEE Mediterranean Conf. on Control and Automation*, Crete, Greece, June 25-28, 2013, pp. 444–451

[15] L. D'Alfonso, E. Garone, P. Muraca, P. Pugliese, On the use of the inclinometers in the PnP Problem, *Proc. 12th European Control Conf.*, Zurich, Switzerland, July 17-19, 2013, pp. 4112–4117

[16] P. Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*, Springer Berlin Heidelberg, 2011

[17] www.logitech.com

[18] http://code.google.com/p/ardu-imu/wiki/HomePage

[19] http://www.vision.caltech.edu/bouguetj/calib_doc

[20] http://petercorke.com/Machine_Vision_Toolbox.html

[21] J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions, *SIAM J. Optimiz.*, 9(1):112–147, 1998