

## On the use of the inclinometers in the $PnP$ Problem

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

**Abstract**—This paper deals with the problem of estimating the relative pose between a camera and an object. It is assumed that both the camera and the object are equipped with an Inertial Measurement Unit (IMU) which measures their inclinations with respect to the gravity vector. Moreover, it is assumed that the object contains a feature of  $n \geq 2$  points, the position of which in the object reference frame is known *a priori*. The resulting pose estimation problem can be seen as a  $PnP$  problem with inclination information. Early results of the authors on the subject showed that in this case the P2P problem always gives 2 solutions, except for a few singular configurations where the number of solution is infinite. Moreover, to avoid singular configurations and to resolve ambiguities a very simple solution for the P3P problem, based on the idea of re-projection, was proposed. In this paper, it will be shown that, thanks to a simple test based on geometrical considerations, very often one of two solutions of the P2P problem can be discarded. Moreover to solve the remaining ambiguities and to ameliorate the pose estimation, a novel and more robust algorithm for the general  $PnP$  problem will be proposed. The results will be validated through numerical and experimental tests.

### 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] and since then has received considerable attention. Fischler and Bolles summarize the problem as follows:

*Given the relative spatial locations of  $n$  control points, and given the angle to every pair of control points from an additional point called the Center of Perspective ( $C_P$ ), find the lengths of the line segments joining  $C_P$  to each of the control points.*

In other words, the problem is the one 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 composed of  $n$  points placed on the object. This problem finds application in many different fields, such as computer vision [7], computer animation [6], automation, image analysis, photogrammetry [8] and robotics [9] [10] [11].

Several solutions to the  $PnP$  problem have been studied 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  $n = 3$ , since the P2P problem ( $n = 2$ ), in its classical formulation, has infinite solutions. Moreover, as proved in [4], in order to have

a unique solution to the  $PnP$  problem, the smallest number of feature points is  $n = 4$ . These points have to be coplanar and no more than two of them have to lie on any single line. In [2] a complete analysis of the P3P problem is provided. There the authors show that this problem has at most four solutions and it can have one solution only in some particular configurations.

For what concerns the P4P problem, several approaches have been presented in the literature. An important result is the one presented by Rivers et al [3] proposing an approach based on the solutions of a set of six quadratic equations with four unknowns. Another important approach is the RANSAC algorithm, introduced by Fischer and Bolles [1], which faces the problem by solving the P3P problem for any groups of 3 points of the feature and then makes the intersection of their solutions.

Note that in the classical  $PnP$  problem all the information is provided by the camera and by the feature. However in many cases, such as in robotics applications, other sensors are available and could provide useful information to obtain a more reliable pose estimation. For instance, mobile robots are usually equipped with Inertial Measurement Units (IMUs) able to measure, in a static context, the gravity vector.

In this paper the idea is to use the inclination information measured by the IMUs in order to help the vision system to solve the  $PnP$  problem. Due to the high resolution of the IMUs' accelerometers this approach is expected not only to simplify the solution of the  $PnP$  problem, but also to give a more accurate pose estimation. To the best of our knowledge, only a few recent works making use of this philosophy have been presented in the literature. Namely, in [5], the authors make use of the roll, pitch and yaw provided by the accelerometers and the magnetometers of an IMU placed on the observed object to compute the translation vector between the feature and a fixed camera with known position and orientation. In the same paper it is proved 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 should be noticed that the attitude computed using both the accelerometers and the compass the latter of which can be quite noisy and unreliable. In [12] the authors assume to know the coordinates of two points in the absolute reference frame and, making use of inclination provided by an IMU mounted on the camera, they solve a P2P problem to reconstruct the absolute camera pose.

In this 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 inclinometers. In this

E. Garone is with Faculty of Applied Science, Control and Systems Analysis Department, Université Libre de Bruxelles, 1050 Brussels, Belgium. E-mail: [egarone@ulb.ac.be](mailto: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](mailto:ldalfonso,muraca,pugliese@dimes.unical.it)

configuration an algorithm able to estimate the relative pose of the object is proposed. Early theoretical results along this line can be found in [17]. There it has been shown that such a modified P2P problem always gives 2 solutions, except when the 2 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. The ambiguities were solved by means of a simple selection process where a P2P problem over two points was first solved and one of the two solutions was chosen by looking at the re-projection of the third point. Starting from this early result, in this paper we will first present a geometrical test that, in roughly an half of the cases, allows one to obtain a unique solution for the P2P Problem. Then, in order to cope with singular configurations and to resolve the remaining ambiguities, a new algorithm able to deal with the general  $PnP$  problem is introduced. This algorithm is based on the intersection&interpolation idea which allows one to improve the obtained estimation as  $n$  grows. The effectiveness of the proposed approach will be shown through a set of numerical and experimental tests. The paper is organized as follows: in Section II we state the problem and define our framework; in Section III we present a parametrization of the rotation matrix between the camera reference frame and the object reference frame; in Section IV the solution to the P2P problem is presented and analyzed; in Section V the solution to the  $PnP$  problem is presented and analyzed; in Section VI experimental results are shown and in Section VII we state out our conclusions.

## II. PROBLEM STATEMENT

Assume a camera and an object in the field of view of the camera. We define two reference frames: the camera reference frame  $O_{xyz}$ , the origin of which is 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 in the object reference frame are known *a priori*. The object and the camera are equipped with an IMU capable of measuring the gravity unit vector:  $\hat{g}_{obj} = [g_u, g_v, g_w]$  in the object reference frame, and  $\hat{g}_{cam} = [g_x, g_y, g_z]$  in the camera reference frame. Please note that, hereafter it is assumed that the distances between the camera and the object are negligible w.r.t. the earth radius and then that  $\hat{g}_{cam}$  and  $\hat{g}_{obj}$  represent the same vector expressed in two different coordinate frames. Moreover we suppose the camera has no distortion and is characterized by the focal length  $f$  and the distance per pixel  $dpx$ . The image reference frame is  $O^c_{x_c y_c}$ , the image plane is  $z = f$  and the image center is  $C_I = (x_{C_I}, y_{C_I})$ . The overall scenario is depicted in Figure 1. Goal of this paper is to make use of the information provided by the camera and of the measurement of the gravity vector in the two reference frames to obtain the transformation matrix between the camera and the object reference frames.

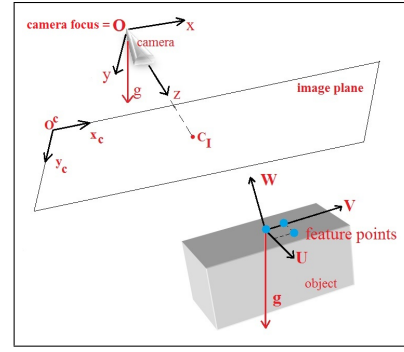


Fig. 1. Camera, Image and Object reference frames

## III. P2P PROBLEM WITH KNOWN VERTICAL DIRECTION

For the reader's convenience, some of the results presented in [17] for the case  $n = 2$  are recalled in the next two subsections.

### A. Rotation Matrix Parameterization

Solving the  $PnP$  problem consists of determining the transformation matrix

$$R_t = \begin{bmatrix} R & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where  $R$  is the rotation matrix and  $t = [t_x, t_y, t_z]^T$  is the translation vector between the object reference frame and the camera reference frame. The first step to find this matrix is to find the rotation matrix  $R$ .

If we knew a certain reference frame  $w$  for which we had the rotation matrix  $R_c^w$  from this reference frame to the camera reference frame, and the rotation matrix  $R_o^w$  from the  $w$  reference frame to the object reference frame, the rotation matrix  $R$  could have been obtained as

$$R = R_c^o = R_c^w [R_o^w]^T \quad (2)$$

The above relationship is true whatever is the  $w$  reference frame. In this case, it is convenient to define an “intermediate” reference frame  $w$  defined as an artificial NED reference frame where the Down-vector is given by the gravity vector and the North-vector (and consequently the East-vector) are chosen in an arbitrary way by choosing a “fake” magnetic unit vector  $\hat{m}_{cam}$  into the camera reference frame. This unit vector has to lie on the plane orthogonal to the gravity unit vector  $\hat{g}_{cam}$ . For simplicity the following unit vector is chosen<sup>1</sup>:

$$\hat{m}_{cam} = \left[ \frac{g_z}{\sqrt{g_x^2 + g_z^2}}, 0, \frac{-g_x}{\sqrt{g_x^2 + g_z^2}} \right]^T \quad (3)$$

The first and the second columns of  $R_c^w$  will be  $\hat{g}_{cam}$  and  $\hat{m}_{cam}$  respectively. The third column of  $R_c^w$  will be simply

$$\hat{n}_{cam} = \hat{g}_{cam} \times \hat{m}_{cam} = \left[ \frac{-g_x g_y}{\sqrt{g_x^2 + g_z^2}}, \sqrt{g_x^2 + g_z^2}, \frac{-g_y g_z}{\sqrt{g_x^2 + g_z^2}} \right]^T \quad (4)$$

<sup>1</sup>Please note that this choice is completely arbitrary. If  $g_x = g_z = 0$ , which implies that the gravity unit vector lies on the  $Y$ -axis, we can always choose another unit vector orthogonal to  $\hat{g}_{cam}$ . By following the same line described in the paper, the formulas to be used to solve the problems in this case can be obtained. Details are omitted for space constraints.

Therefore the rotation matrix from our artificial NED reference frame to the camera reference frame is

$$R_c^w = \begin{bmatrix} \hat{g}_{cam} & \hat{m}_{cam} & \hat{n}_{cam} \end{bmatrix} \quad (5)$$

Following the same lines, it is possible to build the rotation matrix  $R_o^w$  from the  $w$  reference frame to the object reference frame. Clearly in this case the second column should contain  $\hat{m}_{obj}$ , that is our “fake” magnetic unit vector in the object reference frame and the orientation of which on the plane orthogonal to the gravity is unknown. Since this unit vector has to lie on the intersection between the plane orthogonal to the gravity unit vector  $\hat{g}_{obj}$  and a unit sphere, it can be parametrized w.r.t. an unknown angle  $\alpha$

$$\hat{m}_{obj} = \hat{m}_{obj}(\alpha) = \hat{m}_1 \sin \alpha + \hat{m}_2 \cos \alpha \quad (6)$$

where  $\{\hat{m}_1, \hat{m}_2\}$  is an orthonormal basis for the plane orthogonal to  $\hat{g}_{obj}$  and  $\alpha$  characterizes the “fake” magnetic vector orientation. For simplicity we choose  $\hat{m}_1$  as<sup>2</sup>

$$\hat{m}_1 = \left[ \frac{g_w}{\sqrt{g_u^2 + g_w^2}}, 0, \frac{-g_u}{\sqrt{g_u^2 + g_w^2}} \right]^T$$

and thus  $\hat{m}_2$  as

$$\hat{m}_2 = \hat{g}_{obj} \times \hat{m}_1 = \left[ \frac{-g_u g_w}{\sqrt{g_u^2 + g_w^2}}, \sqrt{g_u^2 + g_w^2}, \frac{-g_u g_w}{\sqrt{g_u^2 + g_w^2}} \right]^T$$

The third vector  $\hat{n}_{obj}(\alpha)$  is

$$\hat{n}_{obj} = \hat{n}_{obj}(\alpha) = \hat{g}_{obj} \times \hat{m}_{obj}(\alpha) = \left[ \frac{-g_w \cos \alpha + g_u g_w \sin \alpha}{\sqrt{g_u^2 + g_w^2}}, \sqrt{g_u^2 + g_w^2} \sin \alpha, \frac{g_u \cos \alpha - g_u g_w \sin \alpha}{\sqrt{g_u^2 + g_w^2}} \right]^T \quad (7)$$

Finally, the rotation matrix from our artificial NED reference frame to the object reference frame is

$$R_o^w = R_o^w(\alpha) = \begin{bmatrix} \hat{g}_{obj} & \hat{m}_{obj}(\alpha) & \hat{n}_{obj}(\alpha) \end{bmatrix} \quad (8)$$

and the rotation matrix  $R_c^o = R_c^o(\alpha)$  is:

$$R_c^o = R_c^o(\alpha) = R_c^w [R_o^w(\alpha)]^T \quad (9)$$

where  $\alpha$  is one of the unknowns of the problem.

### B. P2P problem with known vertical direction

The P2P problem corresponds to the case we want to estimate the relative pose of the object in the camera reference frame when the object contains a feature of two distinct points the coordinates of which,  $A = (A_u, A_v, A_w)$  and  $B = (B_u, B_v, B_w)$ , in the object reference frame are known<sup>3</sup>. Let  $K$  be a point in the object reference frame. Using the parametrized rotation matrix  $R(\alpha)$ , provided by the equation (9), the coordinates of  $K$  in the camera reference frame can be written as

$$P_K = [K_x, K_y, K_z]^T = R(\alpha)K + t \quad (10)$$

<sup>2</sup>as in the equation (3), if  $g_u = g_w = 0$ , we could choose another unit vector orthogonal to  $\hat{g}_{obj}$  obtaining similar results.

<sup>3</sup>In this paper we assume without loss of generality that all the feature points are on the plain  $W = 0$ . For the P2P and P3P case we can always obtain this condition by means of a transformation.

The coordinates of the pixel associated with  $P_K$  in the image are defined as

$$x_K = \tilde{f} \frac{K_x}{K_z} + x_{C_I} \quad y_K = \tilde{f} \frac{K_y}{K_z} + y_{C_I} \quad (11)$$

where  $\tilde{f} = \frac{f}{d_{px}}$  is the focal length measured in pixels. Using the above equations it is possible to compute the pixels  $P_{X_A} = (x_A, y_A)$  and  $P_{X_B} = (x_B, y_B)$ , in the image plane, in which the points  $A$  and  $B$  will be projected. Defining  $\tilde{x}_A = (x_A - x_{C_I}) \frac{1}{\tilde{f}}$ ,  $\tilde{y}_A = (y_A - y_{C_I}) \frac{1}{\tilde{f}}$ ,  $\tilde{x}_B = (x_B - x_{C_I}) \frac{1}{\tilde{f}}$ ,  $\tilde{y}_B = (y_B - y_{C_I}) \frac{1}{\tilde{f}}$ , from equations (11) and (10) we obtain

$$\begin{aligned} \tilde{x}_A &= \frac{R_{1,1}A_u + R_{1,2}A_v + R_{1,3}A_w + t_x}{R_{3,1}A_u + R_{3,2}A_v + R_{3,3}A_w + t_z} \\ \tilde{y}_A &= \frac{R_{2,1}A_u + R_{2,2}A_v + R_{2,3}A_w + t_y}{R_{3,1}A_u + R_{3,2}A_v + R_{3,3}A_w + t_z} \\ \tilde{x}_B &= \frac{R_{1,1}B_u + R_{1,2}B_v + R_{1,3}B_w + t_x}{R_{3,1}B_u + R_{3,2}B_v + R_{3,3}B_w + t_z} \\ \tilde{y}_B &= \frac{R_{2,1}B_u + R_{2,2}B_v + R_{2,3}B_w + t_y}{R_{3,1}B_u + R_{3,2}B_v + R_{3,3}B_w + t_z} \end{aligned} \quad (12)$$

where  $R_{i,j}$  is the  $(i, j)$  element of the matrix  $R(\alpha)$ .

The approach used to solve the P2P problem is the following. As a first step, the value of  $\alpha$  is computed using the available information. Then, thanks to (9), we obtain the matrix  $R(\alpha)$ , and finally we determine  $t$  using the results introduced in [5]. Using equations (12) it is possible to prove that, under the assumption  $\Delta_y = \tilde{y}_A - \tilde{y}_B \neq 0$  OR  $\Delta_x = \tilde{x}_A - \tilde{x}_B \neq 0$ ,  $\alpha$  is given as the solution of an equation in the form:

$$a \sin \alpha + b \cos \alpha + c = 0 \quad (13)$$

where  $a$ ,  $b$  and  $c$  are scalar constants that can be computed in closed form on the basis of  $A$ ,  $B$ ,  $\tilde{x}_A$ ,  $\tilde{y}_A$ ,  $\tilde{x}_B$ ,  $\tilde{y}_B$ ,  $\hat{g}_{cam}$ ,  $\hat{g}_{obj}$ . For more details on the form of these terms please refer to [17]. Equation (13) can be solved as

$$\alpha = \arccos(-c/M) + \beta \quad (14)$$

where  $M = \sqrt{a^2 + b^2}$ ,  $\beta = \arctan\left(\frac{a}{b}\right)$ . Please note that equation (14) has two possible solutions, we will denote hereafter as  $\alpha_i, i = 1, 2$ . Once  $\alpha$ , and therefore the rotation matrix  $R(\alpha)$ , is known we can obtain the translation vector  $t$  using the equations presented in [5], here reported for the reader's convenience:

$$\begin{aligned} t_x &= A_z \tilde{x}_A - [R_{1,1}, R_{1,2}, R_{1,3}]A \\ t_y &= A_z \tilde{y}_A - [R_{2,1}, R_{2,2}, R_{2,3}]A \\ t_z &= A_z - [R_{3,1}, R_{3,2}, R_{3,3}]A \end{aligned}$$

where  $A_z$  is computed as follows:

$$\begin{aligned} \text{if } \tilde{x}_B \neq \tilde{x}_A & \\ A_z &= \frac{((R_{1,1}, R_{1,2}, R_{1,3}) - \tilde{x}_B [R_{3,1}, R_{3,2}, R_{3,3}]) (A - B)}{\tilde{x}_A - \tilde{x}_B} \\ \text{otherwise} & \\ A_z &= \frac{((R_{2,1}, R_{2,2}, R_{2,3}) - \tilde{y}_B [R_{3,1}, R_{3,2}, R_{3,3}]) (A - B)}{\tilde{y}_A - \tilde{y}_B} \end{aligned} \quad (15)$$

Given a rotation matrix  $R$  the above equations admit a unique solution for the associated translation vector  $t$ . Then, since there are two possible values for  $\alpha$ ,  $\alpha_1$  and  $\alpha_2$ , two possible rotation matrices,  $R(\alpha_1)$  and  $R(\alpha_2)$  and two translation vectors,  $t_1$  and  $t_2$  result. Please note that, in line of principle, both  $(R(\alpha_1), t_1)$  and  $(R(\alpha_2), t_2)$  are possible solutions for the P2P problem in the case  $\Delta_y \neq 0$  OR  $\Delta_x \neq 0$ . Note that if the above assumption is not satisfied, then the two pixels,  $P_{X_A}$  and  $P_{X_B}$ , are coincident. Note

that in this situation it is not possible to find a finite number of solution to the P2P problem since the two feature points  $A$  and  $B$  are aligned with the camera focus and even if it may possible to reconstruct the attitude of the object, any "depth" information is lost. The following lemma, proved in [17], summarizes the possible cases for the P2P problem:

*Lemma 1: Using information taken from an image, provided by the camera, and on the gravity vectors, provided by the IMUs placed on the camera and on the object, the P2P problem yields to:*

- two solutions for the orientation and the translation when  $\Delta_x \neq 0$  OR  $\Delta_y \neq 0$ ;
- a unique solution<sup>4</sup> for the orientation and an infinite number of solutions for the translation when  $P_{X_A} = P_{X_B}$  and the line between the two feature points and  $O$  is not aligned with the gravity vector ;
- an infinite number of solutions for the orientation and for the translation when  $P_{X_A} = P_{X_B}$  and the line between the two feature points and  $O$  is aligned with the gravity vector.

### C. A test to resolve (some) ambiguity

Goal of this subsection is to define a simple test to resolve some of the ambiguities of the above discussed P2P solutions. In fact, although under the assumption  $\Delta_x \neq 0$  OR  $\Delta_y \neq 0$  the discussed modified P2P problem admits two solutions, in some situations, one of these solutions has not a physical meaning and, therefore, it can be discarded. The main idea is that quite often one of the two computed transformation matrix maps some of the points of the P2P outside the field of view of the camera and then it can be discarded. The camera field of view, as shown in Figure 2, can be defined as the conic combination of the vectors  $v_1, v_2, v_3, v_4$  pointing from the camera focus to the image corners. As a consequence a point  $P$  expressed w.r.t. the camera frame will be in the camera field if and only if it can be represented as a conic combination of  $v_1, v_2, v_3, v_4$ . The latter is equivalent to say that  $P$  can be expressed as a conic combination of either  $v_1, v_2, v_3$  or  $v_1, v_3, v_4$ :

$$\begin{aligned} \exists k_1, k_2, k_3 \geq 0 : [ v_1 \quad v_2 \quad v_3 ] \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} &= P \\ \text{OR} & \\ \exists k_1, k_3, k_4 \geq 0 : [ v_1 \quad v_3 \quad v_4 ] \begin{bmatrix} k_1 \\ k_3 \\ k_4 \end{bmatrix} &= P \end{aligned} \quad (16)$$

Being  $v_1, v_2, v_3$  and  $v_1, v_3, v_4$  two set of 3 conical independent vectors of  $\mathbb{R}^3$ , the latter condition is equivalent to test that:

$$\begin{aligned} [ v_1 \quad v_2 \quad v_3 ]^{-1} P &\geq 0 \\ \text{OR} & \\ [ v_1 \quad v_3 \quad v_4 ]^{-1} P &\geq 0 \end{aligned} \quad (17)$$

where  $\geq 0$  has to be meant element-wise (each entry is bigger than or equal to zero). Note that since the matrices  $[v_1 \ v_2 \ v_3]$  and  $[v_1 \ v_3 \ v_4]$  only depend on the camera characteristics, they do not change in time and their inverse can be computed a priori. Finally, to test if a given transformation matrix  $R_t$  has a physical meaning, the following algorithm can be used

<sup>4</sup>This is true under the assumption it is possible to recognize which of the two point is nearer to the camera, otherwise the possible orientations are two.

---

### Algorithm 1: P2P Solution Feasibility Test

---

- 1 **Compute**  $P_A = RA + t$  and  $P_B = RB + t$
  - 2 **Use** (17) to check if  $P_A$  and the  $P_B$  are in the field of view
  - 3 **if** one of the point is not in the field of view, **discard**  $R_t$
- 

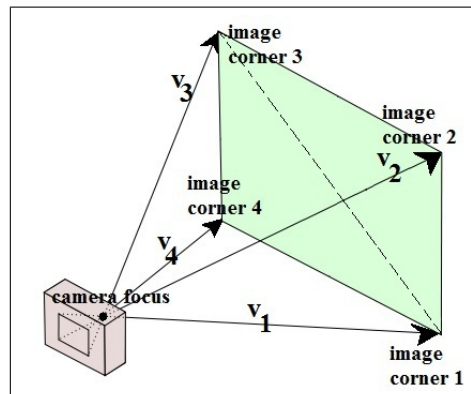


Fig. 2. camera field of view as conic combination

In order to evaluate effectiveness of this test, a set of 125000 randomly generated numerical simulations has been performed. In each test, both the configuration of the object and of the camera were randomly chosen. In the 47% of cases the modified P2P problem have only one solution with a physical meaning.

### IV. PnP PROBLEM WITH KNOWN VERTICAL DIRECTION

In this section we will propose a solution to the stated PnP problem in the general case  $n \geq 3$ . In order to avoid the pathological situation where all the points of the feature  $P_i = (P_{u,i}, P_{v,i}, P_{w,i}), i = 1, \dots, n$  are seen by the camera as a single pixel, we will hereafter assume that at least three points of the feature are not collinear.

In [17] a preliminary algorithm to solve a P3P problem was proposed. There, the idea was to solve the P2P problem on two points and then to choose between the two obtained solutions the one that better mapped the third point in the associated point on the image. It should be remarked that, although theoretically well sounded, in real situations this algorithm can be heavily affected by noise. The main reason is that the information provided by the additional third point is used only to perform a choice but not to improve the estimation.

A possible improvement to the algorithm could be obtained by solving three P2P problems, one for each possible couple of points in the P3P and then to fuse all the solutions to obtain a more reliable solution to the P3P problem. This observation is the basic insight for the algorithm proposed in this paper, the structure of which can be summarized in the following conceptual steps:

**Remark** It is worth to remark that there are many possible ways to implement the line 4 of algorithm 2. A simple

---

**Algorithm 2:** Algorithm for  $PnP$  problem solution

---

- 1 **solve** the P2P problem related to each couple of distinct points (by distinct points we mean points such that are mapped in two different pixels in the image frame) in the set  $\{P_i\}_{i=1}^n$  obtaining  $\binom{n}{2}$  couples of possible solutions  $\{(R(\alpha_{1,i}), t_{1,i}), (R(\alpha_{2,i}), t_{2,i})\}_{i=1}^{\binom{n}{2}}$ .
- 2 **for** any couple  $\{(R(\alpha_{1,i}), t_{1,i}), (R(\alpha_{2,i}), t_{2,i})\}$  **discard** the "wrong" solution and **add** the other one to the set  $\mathcal{S}$
- 3 **whenever possible**, using Algorithm 1.
- 4 **otherwise**, comparing the solutions with the ones in the set  $\mathcal{S}$
- 5 **Interpolate** the couples in  $\mathcal{S}$  as follows

$$\alpha = \frac{1}{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} \alpha_j, \quad t = \frac{1}{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} t_j \quad (18)$$

- 6 The **solution** of the  $PnP$  is  $(R(\alpha), t)$
- 

and effective way is to chose the solution that minimizes some distance w.r.t. the current set  $\mathcal{S}$ . In the experimental tests presented in this paper this is done by choosing the solution that minimizes the distance w.r.t.  $\alpha$  defined as in (18). Note that, in order to avoid initializations problems and to avoid possible singularities, it may be convenient to solve the ambiguity of the first P2P by evaluating the re-projection of a third point as in [17].

**Remark** Some implementation remarks are in order. A first observation is that the algorithm is parallelizable as the  $\binom{n}{2}$  can be computed in parallel and the average can be computed in an incremental way. Moreover it is worth to remark that in the case the number of point is high, it can be convenient not to use all the possible couples of points in  $\{P_i\}_{i=1}^n$  but only some of them. It is however a good policy to never discard points and to choose always at least  $\lceil n/2 \rceil$  couples. Finally note that in order to increase the robustness w.r.t. pixel noise, it may be convenient to substitute (18) with the weighted average

$$\alpha = \frac{1}{\sum_{j=1}^{|\mathcal{S}|} \phi_j} \sum_{j=1}^{|\mathcal{S}|} \phi_j \alpha_j, \quad t = \frac{1}{\sum_{j=1}^{|\mathcal{S}|} \phi_j} \sum_{j=1}^{|\mathcal{S}|} \phi_j t_j \quad (19)$$

where  $\phi_j$  is a function of the distance between the two points into exam in the image frame. In particular, since the bigger this distance the less the estimation is affected by pixel noise,  $\phi_j$  should be used to "weight more" couples with large image distances.

## V. EXPERIMENTAL RESULTS

To test and evaluate the performance of the proposed algorithm for the  $PnP$  problem the following experimental setting has been used:

- A Logitech webcam C310 with resolution  $1280 \times 960$ , [14]. The intrinsic parameters of the camera have been estimated using the Camera Calibration Toolbox, [16].

- Two ArduIMU V3 Inertial Measurement Units [15].
- The four points feature shown in Figure 3. Each square's down left corner is a feature point. The left down corner of the total white square is the point  $P_A = [0 \ 0 \ 0]m$  in the object reference frame. Following the counterclockwise order, the other points are  $P_B = [0.08 \ 0 \ 0]m$ ,  $P_C = [0.08 \ 0.08 \ 0]m$  and  $P_D = [0 \ 0.08 \ 0]m$ .

The  $PnP$  algorithm proposed in this work (Algorithm 2) with  $n = 3$ , denoted by  $P3P_{IMU}$ , has been contrasted with a classical  $P4P$  algorithm [13] based only on the information provided by the camera, and with the early P3P algorithm proposed in [17], denoted by  $P3P_{old}$ .

A set of experiments has been performed using the following procedure

- 1) the camera and the object are placed in an unknown configuration with the object in the field of view of the camera and an estimation  $R_{t,1}^*$  of the actual transformation matrix  $R_{t,1}$  is computed using the 3 algorithms.
- 2) The object is rotated and translated of a known displacement. The new transformation matrix is  $R_{t,2}$ . The relative rotation and translation between  $R_{t,1}$  and  $R_{t,2}$  is the transformation matrix  $R_{t,3}$ .
- 3) An estimation of  $R_{t,2}$ , namely  $R_{t,2}^*$  is obtained.
- 4) An estimation of the displacement matrix between the two configurations is computed as  $R_{t,3}^* = (R_{t,1}^*)^{-1} R_{t,2}^*$ .
- 5) The estimation  $R_{t,3}^* = \begin{bmatrix} R_3^* & t_3^* \\ 0 & 1 \end{bmatrix}$  is compared with  $R_{t,3} = \begin{bmatrix} R_3 & t_3 \\ 0 & 1 \end{bmatrix}$

Figure 3 shows two pictures taken during the experiment. The performances of the above algorithms have been contrasted in terms of displacement error  $\delta t = \|t_3 - t_3^*\|$ , in centimeters, and in terms of error on the rotation matrix  $\delta R = \|R_3 - R_3^*\|$ . Table I and II show the obtained results for different displacements. The first column of the tables reports the vector  $\Delta t = [\Delta t_x, \Delta t_y, \Delta t_z]^T$  representing the translation vector between  $R_{t,1}^*$  and  $R_{t,2}^*$  measured in cm. Since the experiments have been performed on a plane surface, the only angle changed from one experiment to another is the roll angle  $\Delta\phi$ . The second column shows  $\delta t$  and  $\delta R$ , respectively. As shown in the tables, the algorithm  $P3P_{IMU}$  presented in this paper performs in the average better than the other two. In the experiments the interpolation rule (18) has been used. Better results could have been obtained by tuning a more finely tuned interpolation law (19). Finally note that, despite of the fact that only 3 points are used, the two methods making use of the IMU perform in the average much better than the classical P4P solutions. Finally, in order to give a more statistically sounded analysis of the improvements, a set of 1000 randomly generated configurations have been tested. To compare the results the following total reprojection error index has been defined

$$\epsilon = \frac{\epsilon_A + \epsilon_B + \epsilon_C + \epsilon_D}{4} \quad (20)$$

TABLE I  
EXPERIMENTS DATA:  $\delta t$

Displacements	$P4P$	$P3P_{IMU}$	$P3P_{old}$
$\Delta t = [1, 0, 20]^T$	0.04	0.05	0.16
$\Delta t = [-11, 5, 0, 0]^T$	0.22	0.21	0.21
$\Delta t = [-10, 0, 0]^T$	0.23	0.21	0.22
$\Delta t = [-10, 0, 0]^T, \Delta\phi = \frac{\pi}{4}$	0.20	0.20	0.20
$\Delta t = [0, -20, 0]^T$	0.03	0.02	0.03
$\Delta t = [-10, -20, 0]^T$	0.22	0.22	0.23
$\Delta t = [-10, -20, 0]^T, \Delta\phi = \frac{\pi}{4}$	0.22	0.22	0.23
$\Delta t = [-10, -10, 0]^T, \Delta\phi = -\frac{\pi}{4}$	0.20	0.19	0.20
$\Delta t = [0, -10, 0]^T$	0.17	0.02	0.03
$\Delta t = [-20, -10, 0]^T$	0.37	0.39	0.36
$\Delta t = [-20, -10, 0]^T, \Delta\phi = \frac{\pi}{4}$	0.52	0.42	0.43
$\Delta t = [-20, -10, 0]^T, \Delta\phi = -\frac{\pi}{4}$	0.30	0.37	0.39

TABLE II  
EXPERIMENTS DATA:  $\delta R$

Displacements	$P4P$	$P3P_{IMU}$	$P3P_{old}$
$\Delta t = [1, 0, 20]^T$	0.15	0.04	0.04
$\Delta t = [-11, 5, 0, 0]^T$	0.23	0.07	0.60
$\Delta t = [-10, 0, 0]^T$	0.50	0.02	0.07
$\Delta t = [-10, 0, 0]^T, \Delta\phi = \frac{\pi}{4}$	0.22	0.04	0.05
$\Delta t = [0, -20, 0]^T$	0.22	0.04	0.06
$\Delta t = [-10, -20, 0]^T$	0.25	0.04	0.06
$\Delta t = [-10, -20, 0]^T, \Delta\phi = \frac{\pi}{4}$	0.42	0.05	0.06
$\Delta t = [-10, -10, 0]^T, \Delta\phi = -\frac{\pi}{4}$	0.35	0.16	0.19
$\Delta t = [0, -10, 0]^T$	0.28	0.03	0.01
$\Delta t = [-20, -10, 0]^T$	0.43	0.09	0.07
$\Delta t = [-20, -10, 0]^T, \Delta\phi = \frac{\pi}{4}$	0.46	0.08	0.09
$\Delta t = [-20, -10, 0]^T, \Delta\phi = -\frac{\pi}{4}$	0.49	0.07	0.05

where  $\epsilon_q, q \in \{A, B, C, D\}$  is the relative reprojection error on a feature point. For example,  $\epsilon_A$  is  $\epsilon_A = \frac{\|P_{X_A} - P_{X_A}^*\|}{\|P_{X_A}\|}$  and  $P_{X_A}^*$  is the pixel related to the point  $A$  using a computed rotation matrix  $R^*$  and a computed translation vector  $t^*$  obtained with one of the tested algorithms. Simulations show that the averaged reprojection error is 0.0212 for the proposed  $P3P_{IMU}$ , 0.0332 for the  $P3P_{old}$  and 0.1167 for the  $P4P$ .

## VI. CONCLUSIONS

This paper focused on the  $PnP$  problems in a framework where both the camera and the object to be observed are equipped with an IMU providing, in the static context, the information on the gravity vector. In this paper it has been shown that, although the  $P2P$  gives in principle at least 2 solutions, in many cases the ambiguity can be resolved by means of a simple algebraic test. Moreover a new algorithm able to cope with the  $PnP$  problem in the case of known inclination w.r.t. the gravity has been proposed. Such an algorithm presents better performance in terms of scalability and robustness w.r.t. the  $P3P$  problem solution previously presented in [17]. The results presented in this paper have been validated through a series of numerical and experimental tests.

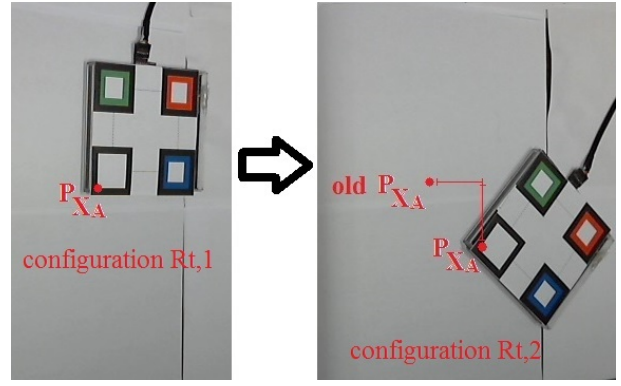


Fig. 3. Pictures taken by the camera: (LEFT) proposed feature in the configuration  $R_{t,1}$ , (RIGHT) proposed feature in the configuration  $R_{t,2}$ . The second picture shows the translation of the pixel  $P_{X_A}$  from its old position (old  $P_{X_A}$ ) to its new position

## 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] X. Gao, X. Hou, J. Tang, H. Cheng, Complete solution classification for the perspective-three-point problem, *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003
- [3] P. Rives, P. Bouthmy, B.Prasada, E.Dubois, Recovering the orientation and the position of a rigid body in space from a single view, *Technical report, INRS-Tlcommunications, 3, place du commerce, Ile-des-Soeurs, Verdun, Quebec, Canada*, 1981
- [4] M. A. Abidi, T. Chandra, A new efficient and direct solution for pose estimation using quadrangular targets; algorithm and evaluation, *IEEE transactions on pattern analysis and machine intelligence*, 17(5):534–538, 1995
- [5] L. Merckel, T. Nishida, Evaluation of a method to solve the perspective-two-point problem using a three-axis orientation sensor, *8th IEEE International Conference on Computer and Information Technology*, 2008
- [6] C. Su, Y. Xu, H. Li, and S. Liu, Application of Wus Method in Computer Animation, *Proc. Fifth Intl Conf. CAD/CG*, 1, 211–215, 1997
- [7] R. Horaud, B. Conio, and O. Le Boulleux, An Analytic Solution for the Perspective 4-Point Problem, *Computer Vision, Graphics, and Image Processing*, 47, 33–44, 1989
- [8] J.S.C. Yuan, A General Photogrammetric Method for Determining Object Position and Orientation, *IEEE Trans. Robotics and Automation*, 5(2), 129–142, 1989
- [9] G. Chen, D. Xu, P. Yang, High Precision Pose Measurement for Humanoid Robot Based on  $PnP$  and  $OI$  Algorithms, *2010 IEEE International Conference on Robotics and Biomimetics*, December 14–18, 2010, Tianjin, China
- [10] Q. Wang, X Zhang, D, Xu, Human Behavior Imitation for a Robot to Play Table Tennis, *Control and Decision Conference (CCDC), 2012 24th Chinese*, 1482 – 1487, 23-25 May 2012
- [11] Q. Wang, X Zhang, D, Xu, On Pose Recovery for Generalized Visual Sensors, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7), 848–861, 2004
- [12] Z. Kukelova, M. Bujnak, T. Pajdla, Closed-form solutions to the minimal absolute pose problems with known vertical direction, *it/Computer vision ACCV 2010, 2011 - Springer*
- [13] -Yubin Hung, Pen-Shu Yeh, David Harwood, Passive ranging to know planar point sets, *IEEE International Conference on Robotics and Automation*, March 1985
- [14] www.logitech.com
- [15] <http://code.google.com/p/ardu-imu/wiki/HomePage>
- [16] [http://www.vision.caltech.edu/bouguetj/calib\\_doc](http://www.vision.caltech.edu/bouguetj/calib_doc)
- [17] L. D’Alfonso, E. Garone, P. Muraca, P. Pugliese,  $P3P$  and  $P2P$  Problems with known camera and object vertical directions, *submitted for the 21st Mediterranean Conference on Control and Automation*, June 25 - 28, Platanias-Chania, Crete, GREECE. Available at <http://www.gprix.it/TN02.pdf>