P3P and P2P Problems with known camera and object vertical directions

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 considered. It is assumed that both the camera and the object are provided with an IMU capable to give their inclinations with respect to the gravity vector. It is moreover assumed that the object contains a feature of 3 points whose position in the object coordinate frame is known. Using the image provided by the camera and the information on the gravity vector by the IMUs we propose an algorithm capable of estimating the relative pose of the object in the camera reference frame by solving a modified P2P or P3P problem. It will be shown that the P2P problem always gives 2 solutions, except in a few singular configurations, while the P3P problem usually gives a single solution. The effectiveness of the proposed approach will be shown by contrasting it with other algorithms presented in the literature.**

I. Introduction

The Perspective-n-Point (PnP) problem, also known as pose estimation, has been introduced 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 exploiting the image provided by the camera and the knowledge of a feature composed of n points placed on the object. Several solutions to the PnP problem have been studied and have found application in many different fields, including computer vision [12], computer animation [11], automation, image analysis, photogrammetry [13] and robotics [14] [15] [16].

From the theoretical viewpoint it has been proved that the smallest number of points which yield to a finite number of solutions for such problem is $n = 3$, since the P2P problem ($n = 2$), in its classical formulation, has infinite solutions.

In [2] a complete analysis of the P3P problem is provided. There the authors show that this problem has at most four solutions and, in some situations, it can have one solution. In [3] a survey on the major direct methods for approaching the P3P problem is provided.

As proved in [6], to have a unique solution to the PnP problem in all object configurations, 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 the literature, the P4P problem has been faced in many ways. Rivers et al [4] propose an approach based on the solutions of a set of six quadratic equations with four unknown. Fischer and Bolles [1] introduced the RANSAC algorithm which faces the problem by solving the P3P problem for any groups of 3 points of the feature and makes use of the intersection of their solutions. In [5] Quan and Lan present a linear algorithm to solve the P4P problem.

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 IMUs (Inertial Measurement Units) able to measure the gravity vector in the object reference frame coordinates through accelerometers and gyros. Usually, IMUs give also information about earth magnetic field by means of a magnetometer. Therefore, the IMU provides all the information on the rotation of the robot 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 quite imprecise since their measurements may be easily affected by local magnetic fields.

In most of the current applications (see e.g. [9],[10]), the information coming from the IMU and from the camera system are elaborated separately and then fused *a posteriori*. For instance in [9] authors present an extended Kalman filter for precisely determining the unknown transformation between a camera and an IMU. In [10] tight coupling of IMU and camera is achieved by an error-state extended Kalman filter (EKF) which uses each visually tracked feature contribute as an individual measurement for its update.

In this paper, on the contrary, the idea is to use the data elaborated from 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 in recent years a few works making use of this philosophy have been proposed. In [8], the authors use the knowledge of roll, pitch and yaw given by an IMU placed on the observed object to compute the translation vector between the feature and a fixed camera with known position and orientation. The authors also prove that if all the attitude information is known, the P2P problem admits a unique solution as long as the two image points are

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*

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

distinct. However it should be noticed that the information about the attitude is computed by the IMU using both the knowledge of the gravity vector and the compass, which, as already remarked, can be quite noisy and unreliable. In [17] the authors assume to know the coordinates of two points in the absolute reference frame and, making use of the roll and pitch provided by an IMU mounted on the camera, solve a P2P problem to reconstruct the absolute camera pose. The yaw angle provided by the IMU is discarded due to the low accuracy of the compass.

In this paper we consider a more general scenario under the assumption that no absolute coordinates of either the camera or the object are known and that both of them are equipped with an IMU providing only their inclination with respect to the gravity vector. In such configuration we propose an algorithm able to estimate the relative pose of the object. We will show that in this configuration the P2P problem always gives 2 solutions, except in a few singular configurations that will be carefully analyzed. We will also show that the P3P problem usually gives a single solution.
This kind of algorithm may be used in several interesting applications. Consider for instance a team of mobile and/or aerial robots, each equipped with an IMU and a camera. Using our algorithm it would be possible for each robot to know the relative position and orientation w.r.t. the others. Such information could be used, for instance, to fuse the camera images so as to have a stereoscopic vision of the world in which the robots move.

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 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 P3P problem is presented and analyzed; in Section VI numerical results are shown and in Section VII we state out our conclusions.

II. Problem Statement

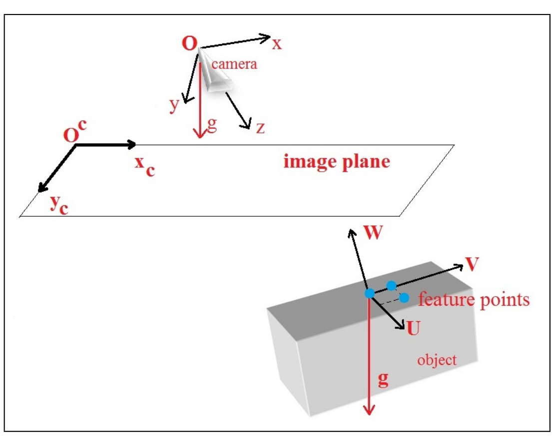

Fig. 1. Camera, Image and Object reference frames

Assume a camera and an object in the angle of view of the camera. We define two reference frames: the camera reference frame, O_{xyz}, and the object reference frame, O'_{UVW}. It is assumed that the object has a feature composed of n points whose coordinates in the object reference frame are known. 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. 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 we assume to know the coordinates of the image center, $C_I = (x_{C_I}, y_{C_I})$. 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 knowledge of the gravity vectors to obtain the transformation matrix between the camera and the object reference frames.

III. Rotation Matrix Parametrization

Solving the PnP problem consists in determining the transformation matrix

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

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 such 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 the camera reference frame to this reference frame, and the rotation matrix R^o_w from the object reference frame to the w reference frame, the rotation matrix R could have been obtained as

$$R = R^o_c = [R^c_w]^T R^o_w \tag{1}$$

The above relationship is true whatever is the w reference frame. As well known, in order to obtain the rotation matrix from a reference frame to another one, we have to represent the canonical basis of the first frame in the second one, assuming no translation. By the IMUs we have information on the gravity unit vector in the object and in the camera reference frames. Therefore may be 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. At this point, using this artificial "fake" NED reference frame, we proceed by finding the rotation matrices R^c_w and R^o_w. Using them we will eventually obtain R by the equation (1).

The first column of R^w_c is the unit vector $\hat{g}_{cam}$. To complete such matrix we arbitrarily choose a "fake" magnetic unit vector $\hat{m}_{cam}$ into the camera reference frame. Such unit vector has to lie on the plane orthogonal to the gravity unit vector $\hat{g}_{cam}$. For simplicity we can chose the following unit vector[1]:

[1]Please note that this choice if 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.

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

which will be the second column of R_c^w. The third column of R_c^w will be simply

$$\hat{n}_{cam} = \hat{g}_{cam} \times \hat{m}_{cam} =$$
$$\left[-\frac{g_x g_y}{\sqrt{g_x^2 + g_z^2}} \ , \ \sqrt{g_x^2 + g_z^2} \ , \ -\frac{g_y g_z}{\sqrt{g_x^2 + g_z^2}} \right]^T \tag{3}$$

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

$$R_c^w = \begin{bmatrix} | & | & | \\ \hat{g}_{cam} & \hat{m}_{cam} & \hat{n}_{cam} \\ | & | & | \end{bmatrix} \tag{4}$$

Since a rotation matrix is an orthonormal matrix, we have R_w^c as $R_w^c = [R_c^w]^T$.

The next step is to build the rotation matrix R_w^o from the object reference frame to the w reference frame. Following the same lines of what done with R_w^c, the first column of R_o^w is $\hat{g}_{obj}$. The second column should contain $\hat{m}_{obj}$, that is our "fake" magnetic unit vector in the object reference frame. However, the coordinates of such vector in the object reference frame are unknown. The only information we have is that it has to lie on the plane orthogonal to $\hat{g}_{obj}$ and it has to be a unit vector, therefore with unit norm. As a consequence, from a geometrical point of view this vector will lie on the intersection between the plane orthogonal to $\hat{g}_{obj}$ and a sphere with unit ray, as shown in Figure 2.

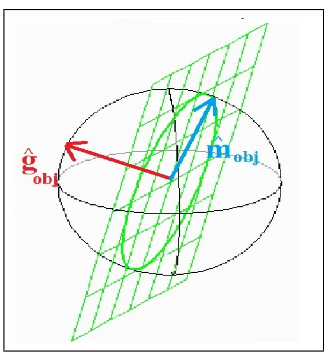

Fig. 2. unit vectors $\hat{g}_{obj}$ and $\hat{m}_{obj}$

This allows us to introduce the following parametrization for $\hat{m}_{obj}$

$$\hat{m}_{obj} = \hat{m}_{obj}(\alpha) = \hat{m}_1 \sin\alpha + \hat{m}_2 \cos\alpha \tag{5}$$

where $\{\hat{m}_1, \hat{m}_2\}$ is an orthonormal basis for the plane orthogonal to $\hat{g}_{obj}$ and α is an unknown angle characterizing the artificial magnetic vector orientation. For simplicity we choose $\hat{m}_1$ as[2]

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

and thus $\hat{m}_2$ as

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

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

$$\hat{n}_{obj} = \hat{n}_{obj}(\alpha) = \hat{g}_{obj} \times \hat{m}_{obj}(\alpha) =$$
$$\left[-\frac{g_w \cos\alpha + g_u g_v \sin\alpha}{\sqrt{g_u^2 + g_w^2}}, \sqrt{g_u^2 + g_w^2} \sin\alpha, \frac{g_u \cos\alpha - g_v g_w \sin\alpha}{\sqrt{g_u^2 + g_w^2}} \right]^T \tag{6}$$

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

$$R_o^w = R_o^w(\alpha) = \begin{bmatrix} | & | & | \\ \hat{g}_{obj} & \hat{m}_{obj}(\alpha) & \hat{n}_{obj}(\alpha) \\ | & | & | \end{bmatrix} \tag{7}$$

At this point using $R_o^w(\alpha)$ and R_c^w, the rotation matrix $R_c^o = R_c^o(\alpha)$ is:

$$R_c^o = R_c^o(\alpha) = R_c^w [R_o^w(\alpha)]^T \tag{8}$$

Where α is one of the unknowns of the problem.

IV. 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 whose coordinates $A = (A_u, A_v, A_w)$ and $B = (B_u, B_v, B_w)$ in the object reference frame are known[3].

Using the parametrized rotation matrix $R(\alpha)$, provided by the equation (8), the coordinates of the points A and B in the camera reference frame can be written as

$$\begin{aligned} P_A &= [A_x, A_y, A_z]^T = R(\alpha)A + t \\ P_B &= [B_x, B_y, B_z]^T = R(\alpha)B + t \end{aligned} \tag{9}$$

In the image plane, the points A and B will be projected into the pixels $P_{X_A} = (x_A, y_A)$ and $P_{X_B} = (x_B, y_B)$ defined as

$$\begin{aligned} x_A &= \tfrac{f}{dpx}\tfrac{A_x}{A_z} + x_{C_I} \quad & y_A &= \tfrac{f}{dpx}\tfrac{A_y}{A_z} + y_{C_I} \\ x_B &= \tfrac{f}{dpx}\tfrac{B_x}{B_z} + x_{C_I} \quad & y_B &= \tfrac{f}{dpx}\tfrac{B_y}{B_z} + y_{C_I} \end{aligned} \tag{10}$$

Defining $\tilde{x}_A = (x_A - x_{C_I})\frac{dpx}{f}, \tilde{y}_A = (y_A - y_{C_I})\frac{dpx}{f}, \tilde{x}_B = (x_B - x_{C_I})\frac{dpx}{f}, \tilde{y}_B = (y_B - y_{C_I})\frac{dpx}{f}$, from equations (10)

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

[3]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.

and (9) we obtain

$$\begin{aligned}
\tilde{x}_A &= \frac{R_{1,1}A_u + R_{1,2}A_v + R_{1,3}A_w + t_x}{R_{3,1}A_u + R_{3,2}A_v + R_{3,3}A_w + t_z} \\
\tilde{y}_A &= \frac{R_{2,1}A_u + R_{2,2}A_v + R_{2,3}A_w + t_y}{R_{3,1}A_u + R_{3,2}A_v + R_{3,3}A_w + t_z} \\
\tilde{x}_B &= \frac{R_{1,1}B_u + R_{1,2}B_v + R_{1,3}B_w + t_x}{R_{3,1}B_u + R_{3,2}B_v + R_{3,3}B_w + t_z} \\
\tilde{y}_B &= \frac{R_{2,1}B_u + R_{2,2}B_v + R_{2,3}B_w + t_y}{R_{3,1}B_u + R_{3,2}B_v + R_{3,3}B_w + t_z}
\end{aligned} \tag{11}$$

where $R_{i,j}$ is the (i,j) element of the matrix $R(\alpha)$.
At this point the goal is to find the α angle. Under the assumption $\Delta_y = \tilde{y}_A - \tilde{y}_B \neq 0$ AND $\Delta_x = \tilde{x}_A - \tilde{x}_B \neq 0$, using equations (11) we can find the following equation:

$$a \sin\alpha + b\cos\alpha + c = 0 \tag{12}$$

where

$$\begin{aligned}
a &= a(A, B, \tilde{x}_A, \tilde{y}_A, \tilde{x}_B, \tilde{y}_B, \hat{g}_{cam}, \hat{g}_{obj}), \\
b &= b(A, B, \tilde{x}_A, \tilde{y}_A, \tilde{x}_B, \tilde{y}_B, \hat{g}_{cam}, \hat{g}_{obj}), \\
c &= c(A, B, \tilde{x}_A, \tilde{y}_A, \tilde{x}_B, \tilde{y}_B, \hat{g}_{cam}, \hat{g}_{obj})
\end{aligned}$$

are scalars constants that can be computed in closed form on the basis of the measured data (the details can be found in the Appendix I). Equation (12) can be solved using the relation

$$a\sin\alpha + b\cos\alpha = M\cos(\alpha - \beta)$$

where

$$M = \sqrt{a^2 + b^2}\ ,\ \ \beta = \arctan\left(\frac{a}{b}\right)$$

Finally we obtain:

$$\alpha = \arccos\left(-\frac{c}{M}\right) + \beta \tag{13}$$

Note that equation (13) has two possible solutions, we will denote hereafter as $\alpha_i, i = 1, 2$.

Once we have the α angle and, therefore, the rotation matrix $R(\alpha)$, we can obtain the translation vector t using the equations presented in [8] which are here reported for the reader 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:
if $\tilde{x}_B \neq \tilde{x}_A$

$$A_z = \frac{([R_{1,1}, R_{1,2}, R_{1,3}] - \tilde{x}_B[R_{3,1}, R_{3,2}, R_{3,3}])(A - B)}{\tilde{x}_B - \tilde{x}_A}$$

otherwise

$$A_z = \frac{([R_{2,1}, R_{2,2}, R_{2,3}] - \tilde{y}_B[R_{3,1}, R_{3,2}, R_{3,3}])(A - B)}{\tilde{y}_B - \tilde{y}_A} \tag{14}$$

Given a rotation matrix R the equations above have a unique solution for the associated translation vector t. Then, since we have two possible values for α, α_1 and α_2, we have two possible rotation matrices, R_1 and R_2, respectively, and for each of these matrices a translation vector, t_1 and t_2.
The couples (R_1, t_1) and (R_2, t_2) are both possible solutions for the P2P problem in the case $\Delta_y \neq 0$ AND $\Delta_x \neq 0$.

Clearly there are three pixel configurations in which the assumption $\Delta_y \neq 0$ AND $\Delta_x \neq 0$ is not true that we have to analyze separately:

1) $\Delta_x = 0$ AND $\Delta_y \neq 0$
2) $\Delta_x \neq 0$ AND $\Delta_y = 0$
3) $\Delta_x = 0$ AND $\Delta_y = 0$

The first configuration may be solved using the first and the third equation in (11) that for $\tilde{x}_A = \tilde{x}_B$ allows us to obtain

$$a_1\sin\alpha + b_1\cos\alpha + c_1 = 0 \tag{15}$$

where

$$\begin{aligned}
a_1 &= a_1(A, B, \tilde{x}_A, \hat{g}_{cam}, \hat{g}_{obj}), \\
b_1 &= b_1(A, B, \tilde{x}_A, \hat{g}_{cam}, \hat{g}_{obj}), \\
c_1 &= c_1(A, B, \tilde{x}_A, \hat{g}_{cam}, \hat{g}_{obj})
\end{aligned}$$

are scalar constants that can be computed in closed form (see Appendix I).
Similarly, the second configuration may be faced using the second and the fourth equation in (11) that for $\tilde{y}_A = \tilde{y}_B$ give

$$a_2\sin\alpha + b_2\cos\alpha + c_2 = 0 \tag{16}$$

where

$$\begin{aligned}
a_2 &= a_2(A, B, \tilde{y}_A, \hat{g}_{cam}, \hat{g}_{obj}), \\
b_2 &= b_2(A, B, \tilde{y}_A, \hat{g}_{cam}, \hat{g}_{obj}), \\
c_2 &= c_2(A, B, \tilde{y}_A, \hat{g}_{cam}, \hat{g}_{obj})
\end{aligned}$$

are scalar constants that can be computed in closed form (see Appendix I).
In these two configurations there are two possible solutions for α and then two possible solutions for the P2P problem.
The third pixel configuration is shown in Figure 3

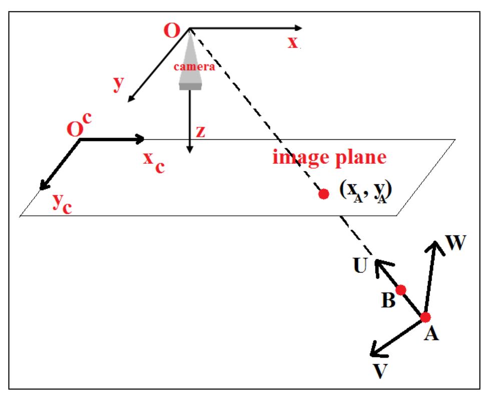

Fig. 3. P2P, $\Delta_x = 0$ AND $\Delta_y = 0$

and corresponds to the case when the two feature points and the origin of the camera reference frame lie on the same line. By using this information we can infer the line unit vector. Depending on the verse of this unit vector there are two possible values for it. However, if we assume to be able to distinguish if the pixel seen in the image is P_{X_A} or P_{X_B} (for example using different colors for the feature points), we can find the correct unit vector. At this point if the line unit vector is not aligned with the gravity unit vector, then it is possible to infer the α angle using information on the two unit vectors. As a consequence there is only one solution for the rotation matrix. Otherwise if such unit vectors are aligned, as shown in Figure 4, then information on the line unit vector is the same provided by the IMUs and it is not possible to find a unique solution for the orientation.

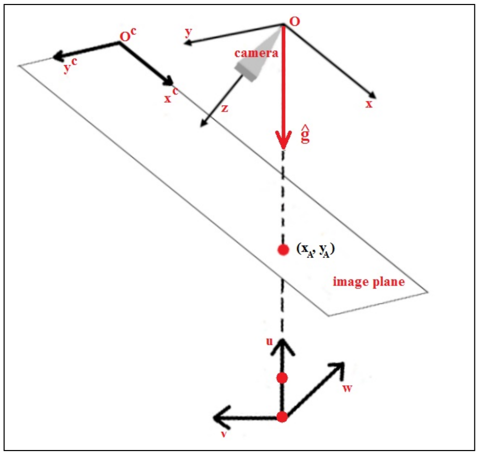

Fig. 4. P3P, $\Delta_x = 0$ AND $\Delta_y = 0$: line unit vector aligned with gravity unit vector

In this configuration there are ∞^1 possible solutions for the rotation matrix. It should be noticed that if $\Delta_x = 0$ AND $\Delta_y = 0$ equations (14) cannot be used for the translation. In fact, all the points on the line between the origin of the camera reference frame and the two feature points are possible values for t. In other words, in such configuration any possibility to infer the distance between the two reference frames is lost and then there are ∞^1 solutions for the translation.

Thanks to the above analysis we can finally state the following lemma:

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 for the orientation and an infinite number of solutions for the translation when $\Delta_x = 0$ AND $\Delta_y = 0$ 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 $\Delta_x = 0$ AND $\Delta_y = 0$ and the line between the two feature points and O is aligned with the gravity vector.*

In conclusion, the P2P problem solved with a camera and two IMUs, always gives 2 solutions, except in a few singular configurations that were analyzed. Please note that with the *classical* approach (using only the camera), this problem always gives an infinite number of solutions [1].

V. P3P PROBLEM WITH KNOWN VERTICAL DIRECTION

The P3P problem corresponds to the case we want to estimate in the camera reference frame the relative pose of the object when it contains a feature of three distinct points whose coordinates $A = (A_u, A_v, A_w)$, $B = (B_u, B_v, B_w)$ and $C = (C_u, C_v, C_w)$, in the object reference frame are known. The only assumption on these three points is that they have not to be collinear. As in the P2P problem, we are interested in finding the rotation matrix R and the translation vector t between the object reference frame and the camera reference frame.

The first main difference with respect to the P2P case is that, being the three points non collinear, it is not possible to see a unique pixel in the image provided by the camera since it is impossible to have the three points and the origin of the camera reference frame on the same line.

Our approach to this problem is based on the results obtained on the P2P problem. The main idea is

1) to solve the P2P problem related to the points A and B;
2) to use the third point, C, and its related pixel in the image, P_{X_C}, to choose one of the solutions of the P3P problem between the solutions of the P2P problem between A and B.

More precisely, assuming $\Delta_x \neq 0$ OR $\Delta_y \neq 0$, we first solve a P2P problem with points A, B and pixels P_{X_A}, P_{X_B} by using the method presented in the previous Section. By doing so we obtain two possible solutions, (R_1, t_1) and (R_2, t_2), and we choose the one which better project the point C into the pixel P_{X_C}. More formally, using equations (10), we compute the pixels $P_{X_C,1}$ and $P_{X_C,2}$ which project the point C using (R_1, t_1) and (R_2, t_2) respectively. If $||P_{X_C,1} - P_{X_C}|| < ||P_{X_C,2} - P_{X_C}||$ then we choose (R_1, t_1) as the solution of the P3P problem, otherwise we choose (R_2, t_2).

If the assumption $\Delta_x \neq 0$ OR $\Delta_y \neq 0$ is not verified, then two of the feature points, namely A and B, are aligned with the point O, as shown in Figure 5.

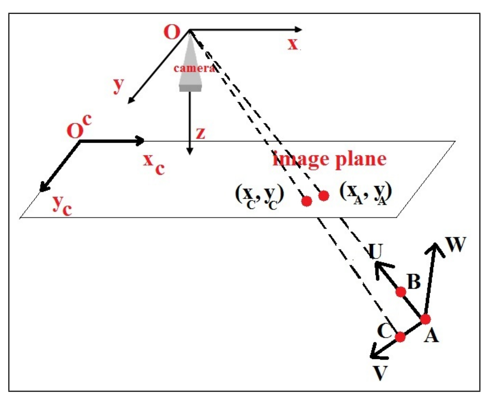

Fig. 5. P3P, $\Delta_x = 0$ AND $\Delta_y = 0$

In such configuration $\Delta_x = 0$ AND $\Delta_y = 0$ and two line unit vectors can be defined, one related to the line between A, B and O and one related to the line between O and C. Since we suppose the three feature points are not on the same line, these unit vectors cannot be aligned. As a consequence, even if one of the two line unit vectors is aligned with the gravity unit vector, it is always possible to use the other one along with the gravity unit vector to infer the angle α. Since the third pixel, P_{X_C} cannot lie on the same line of A, B and O, using $A \equiv B$ and C the translation vector can be determined by suitably adapting equation (14).

Remark Please note that compared to other algorithms based on the solution of the P3P and P4P problems, our algorithm is typically faster since it is based on the solution of a simple P2P problem and of a few further tests to resolve the ambiguity between its 2 solutions. Finally note that in order to increase the robustness w.r.t. pixel noise, it may be convenient to perform the P2P on the couple of point showing the maximal distance in the image plane.

VI. Numerical Results

To evaluate the performance of the proposed solution to the P3P problem we choose the following coordinates, in meters, for the feature points in the object reference frame: $A = (0, 0, 0)$, $B = (0.1, 0.1, 0)$, $C = (0.1, 0, 0)$ and we perform two sets of tests.

In the first set of tests, we used nominal values for all the measurements and we tested the proposed solution to the P3P problem in about 10^7 randomly generated positions and orientations. In all these tests we did not detect any numerical problem in finding the correct rotation and translation.

In the second set of tests we contrasted the P3P algorithm proposed in this work, denoted by $P3P_{IMU}$, with two algorithms presented in literature:

1) a P3P algorithm based on [8]. This algorithms uses the full orientation matrix given by the 2 IMUs and computes a translation vector using (14) for each couple of points: (A, B), (B, C) and (C, A). The three translation vectors are then averaged. We denote this algorithm by $P3P_{NM}$
2) a classical P4P algorithm [18] based only on the information provided by the camera. This algorithm uses a fourth feature point, $D = (0, 0.1, 0)$, and is denoted by $P4P$.

We introduced some noise on the data provided by the IMUs, on the image and on the measurements of the compass used by $P3P_{NM}$ algorithm. We assumed Gaussian noises with zero mean and standard deviation of:

- $\sigma_g = I_3 * 0.01m$ for g_{cam} and g_{obj}
- $\sigma_{pixel} = 2pixels$ in the image
- $\sigma_{com} = 4°$ for the heading provided by the compass. Please note that this is a very optimistic assumption for the noise on the magnetometer since usually measurements provided by such devices are more noisy.

Such values are consistent with the tests proposed in [17] and [8]. 8000 tests were performed changing object position and orientation randomly in a cube with a side of $5m$.

To evaluate the performance of the above algorithms we defined the following index:

$$\epsilon = \frac{\epsilon_A + \epsilon_B + \epsilon_C + \epsilon_D}{4} \tag{17}$$

TABLE I

Percent of reprojection error occurrences on 8000 tests

Algorithm	$\epsilon < 0.02$	$\epsilon < 0.05$	$\epsilon < 0.15$	$\epsilon < 0.3$
$P3P_{IMU}$	93%	99%	100%	100%
$P3P_{NM}$	92%	98%	99%	100%
$P4P$	56%	81%	95%	98%

where $\epsilon_q, q \in \{A, B, C, D\}$ is the relative reprojection error on a feature point. For example, ϵ_A is

$$\epsilon_A = \frac{||P_{X_A} - P^*_{X_A}||}{||P_{X_A}||}$$

and $P^*_{X_A}$ is defined as

$$x^*_A = \frac{f}{dpx}\frac{[R^*(1{,}1)\;\; R^*(1{,}2)\;\; R^*(1{,}3)]A + t^*_x}{[R^*(3{,}1)\;\; R^*(3{,}2)\;\; R^*(3{,}3)]A + t^*_z} + x_{C_I}$$

$$y^*_A = \frac{f}{dpx}\frac{[R^*(2{,}1)\;\; R^*(2{,}2)\;\; R^*(2{,}3)]A + t^*_y}{[R^*(3{,}1)\;\; R^*(3{,}2)\;\; R^*(3{,}3)]A + t^*_z} + y_{C_I}$$

$$P^*_{X_A} = [x^*_A\;\, y^*_A]^T$$

That 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 above algorithms.

In Figure 6 we show ϵ for some of the tests performed using $P3P_{IMU}$, $P3P_{NM}$ and $P4P$ algorithms, respectively.

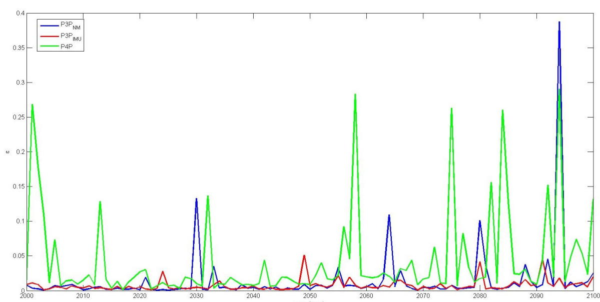

Fig. 6. ϵ in some of the tests performed

As can be seen, the algorithm presented in this paper performs quite well when compared to the other two. The averaged reprojection error on 8000 tests is

- 0.008 using the $P3P_{IMU}$ algorithm
- 0.012 using the $P3P_{NM}$ algorithm
- 0.045 using the $P4P$ algorithm

Please note that the $P3P_{IMU}$ algorithm presents an error one order of magnitude lower than the ones generated by the other two. Table I shows the percent of occurrences, on 8000 tests, of the reprojection error within chosen thresholds. The results presented in the table show that the $P3P_{NM}$ algorithm presents the 92% of reprojection error occurrences under the threshold 0.02 and the $P4P$ algorithm the 56% while our $P3P_{IMU}$ algorithm the 93%.

VII. Conclusions

In this paper we presented an algorithm to solve the P2P and P3P problems in a framework with a camera and an object relatively rotated and translated to each other. To solve such problems we used the image provided by the camera and the information on the gravity vector provided by two IMUs, one placed on the camera and one on the object. We proposed an algorithm able to estimate the relative pose of the object in the camera reference frame by solving a P2P or a P3P problem. It has been shown that with this approach the P2P problem always gives 2 solutions, except in a few singular configurations carefully analyzed while the P3P problem usually gives a single solution. Numerical results prove the effectiveness of the proposed method when contrasted to other P3P and P4P algorithms.

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] R.M. Haralick, The three point perspective pose estimation problem, *Internal note, dept of Elect. Eng., Univ. of Washington*,

[4] P. Rives, P. Bouthmy, B.Prasada, E.Dubois, Recovering the orientation and the position of a rigid body in space from a single view, *Tecnical report, INRS-Tlcommunications, 3, place du commerce, Ile-des-Soeurs, Verdun, Quebec, Canada*, 1981

[5] L. Quan, Z. Lan, Linear N-point camera pose determination, *IEEE transaction on PAMI*, 21(8):774–780, 1999

[6] 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

[7] X. Gao, J. Tang, On the number of solutions of the P4P problem, *MM Research Preprints, MMRC, AMSS, Academis, Sinica, Beijing*, 21:64–76, 2002

[8] 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

[9] F. M. Mirzaei, S.I. Roumeliotis, A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation, *IEEE TRANSACTIONS ON ROBOTICS*, 24(5):1143–1156, 2008

[10] T. Oskiper, S. Samarasekera, R. Kumar, Tightly-Coupled Robust Vision Aided Inertial Navigation Algorithm for Augmented Reality Using Monocular Camera and IMU, *10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland* , October 26-29, 2011.

[11] 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

[12] R. Horaud, B. Conio, and O. Leboulleux, An Analytic Solution for the Perspective 4-Point Problem, *Computer Vision, Graphics, and Image Processing* , 47, 33–44,1989

[13] J.S.C. Yuan, A General Photogrammetric Method for Determining Object Position and Orientation, *IEEE Trans. Robotics and Automation*, 5(2), 129–142, 1989

[14] 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

[15] 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

[16] 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

[17] Z. Kukelova, M. Bujnak, T. Pajdla, Closed-form solutions to the minimal absolute pose problems with known vertical direction, it/Computer visionACCV 2010, 2011 - Springer

[18] Yubin Hung, Pen-Shu Yeh, David Harwood, Passive ranging to know planar point sets, *IEEE International Conference on Robotics and Automation*, March 1985

APPENDIX I

Equation (12)

Assuming $\tilde{x}_A - \tilde{x}_B \neq 0$ AND $\tilde{y}_A - \tilde{y}_B \neq 0$, from the first and the third equation in (11) t_x and t_z may be written as

$$t_z = \frac{1}{(\tilde{x}_A - \tilde{x}_B)\sqrt{(g_u^2+g_w^2)(g_x^2+g_z^2)}}[\sqrt{(g_u^2 + g_w^2)(g_x^2 + g_z^2)}(g_u(-B_u g_x + B_u \tilde{x}_B g_z + A_u(g_x - \tilde{x}_A g_z)) + g_v(-B_v g_x + B_v \tilde{x}_B g_z + A_v(g_x - \tilde{x}_A g_z))) + (g_u g_v(-A_u(g_x \tilde{x}_A + g_z) + B_u(g_x \tilde{x}_B + g_z)) + g_u^2(A_v(g_x \tilde{x}_A + g_z) - B_v(g_x \tilde{x}_B + g_z) + g_w g_y(A_u g_x - B_u g_x - A_u \tilde{x}_A g_z + B_u \tilde{x}_B g_z)) + g_w(A_v g_w(g_x \tilde{x}_A + g_z) - B_v g_w(g_x \tilde{x}_B + g_z) + (g_v^2 + g_w^2) g_y(-B_u g_x + B_u \tilde{x}_B g_z + A_u(g_x - \tilde{x}_A g_z)))) \cos\alpha + ((g_u^2 + g_w^2) g_y(-A_v g_x + B_v g_x + A_v \tilde{x}_A g_z - B_v \tilde{x}_B g_z) + A_u(g_w(g_x \tilde{x}_A + g_z) + g_u g_v g_y(g_x - \tilde{x}_A g_z)) - B_u(g_w(g_x \tilde{x}_B + g_z) + g_u g_v g_y(g_x - \tilde{x}_B g_z))) \sin\alpha]$$

$$t_x = \frac{1}{(\tilde{x}_A - \tilde{x}_B)\sqrt{(g_u^2+g_w^2)(g_x^2+g_z^2)}}[\sqrt{(g_u^2 + g_w^2)(g_x^2 + g_z^2)} (g_u(A_u \tilde{x}_B(g_x - \tilde{x}_A g_z) + B_u(-g_x \tilde{x}_A + \tilde{x}_A \tilde{x}_B g_z)) + g_v(A_v \tilde{x}_B(g_x - \tilde{x}_A g_z) + B_v(-g_x \tilde{x}_A + \tilde{x}_A \tilde{x}_b g_z))) + (g_u g_v(-A_u \tilde{x}_B(g_x \tilde{x}_A + g_z) + B_u \tilde{x}_A(g_x \tilde{x}_B + g_z)) + g_u^2(A_v \tilde{x}_B(g_x \tilde{x}_A + g_z) - B_v \tilde{x}_A(g_x \tilde{x}_B + g_z) + g_w g_y(-B_u g_x \tilde{x}_A + A_u g_x \tilde{x}_B - A_u \tilde{x}_A \tilde{x}_B g_z + B_u \tilde{x}_A \tilde{x}_B g_z)) + g_w(A_v g_w \tilde{x}_B(g_x \tilde{x}_A + g_z) - B_v g_w \tilde{x}_A(g_x \tilde{x}_B + g_z) + (g_v^2 + g_w^2) g_y(A_u \tilde{x}_b(g_x - \tilde{x}_A g_z) + B_u(-g_x \tilde{x}_A + \tilde{x}_A \tilde{x}_B g_z)))) \cos\alpha + (A_u \tilde{x}_B(g_w(g_x \tilde{x}_A + g_z) + g_u g_v g_y(g_x - \tilde{x}_A g_z)) + (g_u^2 + g_w^2) g_y(A_v \tilde{x}_B(-g_x + \tilde{x}_A g_z) + B_v(g_x - \tilde{x}_B g_z)) - B_u \tilde{x}_A(g_w(g_x \tilde{x}_B + g_z) + g_u g_v g_y(g_x - \tilde{x}_b g_z))) \sin\alpha]$$

while from the second and the fourth equation in (11) t_y and t_z are

$$t_z = \frac{1}{(\tilde{y}_A - \tilde{y}_B)\sqrt{(g_u^2+g_w^2)(g_x^2+g_z^2)}}[\sqrt{(g_u^2 + g_w^2)(g_x^2 + g_z^2)}(g_u(-B_u g_y + B_u \tilde{y}_B g_z + A_u(g_y - \tilde{y}_A g_z)) + g_v(-B_v g_y + B_v \tilde{y}_B g_z + A_v(g_y - \tilde{y}_A g_z))) + (g_u g_v g_x(-A_u \tilde{y}_A + B_u \tilde{y}_B) + g_u^2(g_x(A_v \tilde{y}_A - B_v \tilde{y}_B) + B_u g_w(g_x^2 + g_y \tilde{y}_B g_z + g_z^2) - A_u g_w(g_x^2 + g_z(g_y \tilde{y}_A + g_z))) + g_w(g_w g_x(A_v \tilde{y}_A - B_v \tilde{y}_B) - A_u(g_v^2 + g_w^2)(g_x^2 + g_z(g_y \tilde{y}_A + g_z)) + B_u(g_v^2 + g_w^2)(g_x^2 + g_z(g_y \tilde{y}_B + g_z)))) \cos\alpha + (g_w(g_x(A_u \tilde{y}_A - B_u \tilde{y}_B) - B_v g_w(g_x^2 + g_y \tilde{y}_B g_z + g_z^2) + A_v g_w(g_x^2 + g_z(g_y \tilde{y}_A + g_z))) + g_u g_v(-A_u(g_x^2 + g_z(g_y \tilde{y}_A + g_z)) + B_u(g_x^2 + g_z(g_y \tilde{y}_B + g_z))) + g_u^2(A_v(g_x^2 + g_z(g_y \tilde{y}_A + g_z)) - B_v(g_x^2 + g_z(g_y \tilde{y}_B + g_z)))) \sin\alpha]$$

$$t_y = \frac{1}{(\tilde{y}_A - \tilde{y}_B)\sqrt{(g_u^2+g_w^2)(g_x^2+g_z^2)}}[\sqrt{(g_u^2 + g_w^2)(g_x^2 + g_z^2)}(g_u(A_u \tilde{y}_B(g_y - \tilde{y}_A g_z) + B_u(-g_y \tilde{y}_A + \tilde{y}_A \tilde{y}_B g_z)) + g_v(A_v \tilde{y}_B(g_y - \tilde{y}_A g_z) + B_v(-g_y \tilde{y}_A + \tilde{y}_A \tilde{y}_B g_z))) + (g_u(-A_u + B_u) g_v g_x \tilde{y}_A B_v + g_u^2(B_u g_w \tilde{y}_A(g_x^2 + g_z(g_y \tilde{y}_B + g_z)) - \tilde{y}_B((-A_v + B_v) g_w g_x \tilde{y}_A + A_u(g_v^2 + g_w^2)(g_x^2 + g_z(g_y \tilde{y}_A + g_z))))) \cos\alpha + (g_u g_v(-A_u \tilde{y}_B(g_x^2 + g_y \tilde{y}_A g_z + g_z^2) + B_u \tilde{y}_A(g_x^2 + g_y \tilde{y}_B g_z + g_z^2)) + g_u^2(A_v \tilde{y}_B(g_x^2 + g_y \tilde{y}_A g_z + g_z^2) - B_v \tilde{y}_A(g_x^2 + g_y \tilde{y}_B g_z + g_z^2)) + g_w(-B_v g_w \tilde{y}_A(g_x^2 + g_z(g_y \tilde{y}_B + g_z)) + \tilde{y}_B((A_u - B_u) g_x \tilde{y}_A + A_v g_w(g_x^2 + g_y \tilde{y}_A g_z + g_z^2)))) \sin\alpha]$$

By equating the two values of t_z we obtain

$$a \sin\alpha + b \cos\alpha + c = 0$$

where

$$a=\frac{1}{\tilde{x}_A-\tilde{x}_B}[(g_u^2+g_v^2)g_y(-A_vg_x+B_vg_x+A_v\tilde{x}_Ag_z-B_v\tilde{x}_Bg_z)+A_u(g_w(g_x\tilde{x}_A+g_z)+g_ug_vg_y(g_x-\tilde{x}_Ag_z))-B_u(g_w(g_x\tilde{x}_B+g_z)+g_ug_vg_y(g_x-\tilde{x}_Bg_z))]-\frac{1}{\tilde{y}_A-\tilde{y}_B}[g_w(g_x(A_u\tilde{y}_A-B_u\tilde{y}_B)-B_vg_w(g_x^2+g_y\tilde{y}_Bg_z+g_z^2)+A_vg_w(g_x^2+g_z(g_y\tilde{y}_A+g_z)))+g_ug_v(-A_u(g_x^2+g_z(g_y\tilde{y}_A+g_z))+B_v(g_x^2+g_z(g_y\tilde{y}_B+g_z)))+g_u^2(A_v(g_x^2+g_z(g_y\tilde{y}_A+g_z))-B_v(g_x^2+g_z(g_y\tilde{y}_B+g_z)))]$$

$$b=\frac{1}{\tilde{x}_A-\tilde{x}_B}[g_ug_v(-A_u(g_x\tilde{x}_A+g_z)+B_u(g_xB_v+g_z))+g_u^2(A_v(g_x\tilde{x}_A+g_z)-B_v(g_x\tilde{x}_B+g_z)+g_wg_y(A_ug_x-B_ug_x-A_u\tilde{x}_Ag_z+B_u\tilde{x}_Bg_z))+g_w(A_vg_w(g_x\tilde{x}_A+g_z)-B_vg_w(g_x\tilde{x}_B+g_z)+(g_v^2+g_w^2)g_y(-B_ug_x+B_u\tilde{x}_Bg_z+A_u(g_x-\tilde{x}_Ag_z)))]-\frac{1}{\tilde{y}_A-\tilde{y}_B}[g_ug_vg_x(-A_u\tilde{y}_A+B_u\tilde{y}_B)+g_u^2(g_x(A_v\tilde{y}_A-B_v\tilde{y}_B)+B_ug_w(g_x^2+g_y\tilde{y}_Bg_z+g_z^2)-A_u(g_v^2+g_w^2)(g_x^2+g_z(g_y\tilde{y}_A+g_z))+B_u(g_v^2+g_w^2)(g_x^2+g_z(g_y\tilde{y}_B+g_z)))]$$

$$c=\frac{\sqrt{(g_u^2+g_w^2)(g_x^2+g_z^2)}}{\tilde{x}_A-\tilde{x}_B}[g_u(-B_ug_x+B_u\tilde{x}_Bg_z+A_u(g_x-\tilde{x}_Ag_z))+g_v(-B_vg_x+B_v\tilde{x}_Bg_z+A_v(g_x-\tilde{x}_Ag_z))]-\frac{\sqrt{(g_u^2+g_w^2)(g_x^2+g_z^2)}}{\tilde{y}_A-\tilde{y}_B}[g_u(-B_ug_y+B_u\tilde{y}_Bg_z+A_u(g_y-\tilde{y}_Ag_z))+g_v(-B_vg_y+B_v\tilde{y}_Bg_z+A_v(g_y-\tilde{y}_Ag_z))]$$

Configuration $\Delta_x = 0$ *AND* $\Delta_y \neq 0$

$$a_1=-(A_v-B_v)(g_u^2+g_w^2)g_y(g_x-\tilde{x}_Ag_z)+A_u(g_w(g_x\tilde{x}_A+g_z)+g_ug_vg_y(g_x-\tilde{x}_Ag_z))-B_u(g_w(x\tilde{x}_A+g_z)+g_ug_vg_y(g_x-\tilde{x}_Ag_z))$$

$$b_1=-g_u(A_u-B_u)g_v(g_x\tilde{x}_A+g_z)+g_u^2(A_v(g_x\tilde{x}_A+g_z)-B_v(g_x\tilde{x}_A+g_z)+(A_u-B_u)g_wg_y(g_x-\tilde{x}_Ag_z))+g_w(A_vg_w(g_x\tilde{x}_A+g_z)-B_vg_w(g_x\tilde{x}_A+g_z)+(A_u-B_u)(g_v^2+g_w^2)g_y(g_x-\tilde{x}_Ag_z))$$

$$c_1=g_u(A_u-B_u)+g_v(A_v-B_v)(g_x-\tilde{x}_Ag_z)\sqrt{(g_u^2+g_w^2)(g_x^2+g_z^2)}$$

Configuration $\Delta_x \neq 0$ *AND* $\Delta_y = 0$

$$a_2=-g_u(A_u-B_u)g_v(g_x^2+g_z(g_y\tilde{y}_A+g_z))+g_u^2(A_v-B_v)(g_x^2+g_z(g_y\tilde{y}_A+g_z))+g_w((A_u-B_u)g_x\tilde{y}_A-B_vg_w(g_x^2+g_y\tilde{y}_Ag_z+g_z^2)+A_vg_w(g_x^2+g_z(g_y\tilde{y}_A+g_z)))$$

$$b_2=g_u(-A_u+B_u)g_vg_x\tilde{y}_A+g_u^2((A_v-B_v)g_x\tilde{y}_A+B_ug_w(g_x^2+g_y\tilde{y}_Ag_z+g_z^2)-A_ug_w(g_x^2+g_z(g_y\tilde{y}_A+g_z)))+g_w((A_v-B_v)g_wg_x\tilde{y}_A-A_u(g_v^2+g_w^2)(g_x^2+g_z(g_y\tilde{y}_A+g_z))+B_u(g_v^2+g_w^2)(g_x^2+g_z(g_y\tilde{y}_A+g_z)))$$

$$c_2=g_u(A_u-B_u)+g_v(A_v-B_v)(g_y-\tilde{y}_Ag_z)\sqrt{(g_u^2+g_w^2)(g_x^2+g_z^2)}$$