Noname manuscript No.
(will be inserted by the editor)

A minimal closed-form solution for the Perspective Three orthogonal Angles (P3oA) problem. Application to visual odometry

Jesus Briales · Javier Gonzalez-Jimenez

Received: date / Accepted: date

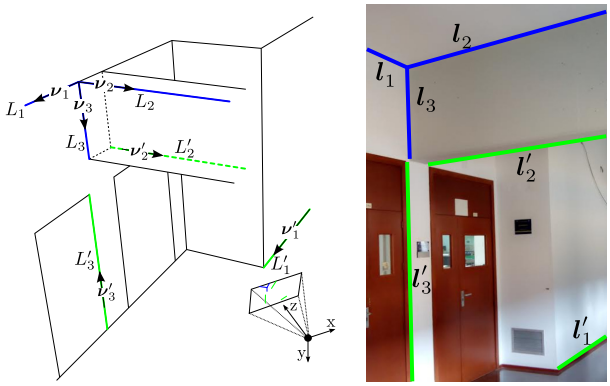
Abstract We provide a simple closed-form solution to the *Perspective Three orthogonal Angles* (P3oA) problem: given the projection of three orthogonal lines in a calibrated camera, find their 3D directions. Upon this solution, an algorithm for the estimation of the camera relative rotation between two frames is proposed. The key idea is to detect triplets of orthogonal lines in a hypothesize-and-test framework and use all of them to compute the camera rotation in a robust way. This approach is suitable for human-made environments where numerous groups of orthogonal lines exist. We evaluate the numerical stability of the P3oA solution and the estimation of the relative rotation with synthetic and real data, comparing our results to other state-of-the-art approaches.

Keywords 2D-3D registration · Minimal solution · Rotation estimation · Visual gyroscope

1 Introduction

The estimation of camera's position and orientation (pose) plays a fundamental role in many applications related to Computer Vision, Robotics, Augmented Reality or Photogrammetry. A great deal of research has been done on 2D-3D registration problems, which consist in finding the relative pose for which 3D geometric features of certain type match their 2D counterparts in image. Several distinct problems exist depending on the nature of the considered features. Some well-known examples are the *inverse perspective problems*:

J. Briales · J. Gonzalez-Jimenez
MAPIR-UMA Group, Dept. Ingeniería de Sistemas y Automática. Universidad de Malaga, 29071 Malaga, Spain
E-mail: jesusbriales@uma.es



(a) 3D scene (b) Camera image

Fig. 1 Examples of P3oA problem in real world. We solve line directions $\boldsymbol{\nu}_k$ for both intersecting $(\mathrm{l}_1, \mathrm{l}_2, \mathrm{l}_3)$ and non intersecting $(\mathrm{l}'_1, \mathrm{l}'_2, \mathrm{l}'_3)$ cases

Perspective-n-Points (PnP), *Perspective-n-Lines* (PnL) and *Perspective-n-Angles* (PnA) problems, where the features involved are points, lines and angles, respectively. Even though some of these problems have been thoroughly studied and numerous solutions proposed (as early as in 1841 for P3P problem [1]) much progress is still being done for PnP [2,3,4], PnL [5,6], or even mixtures of both points and lines [7]. A remarkable aspect of all these problems is that positional data comes into play, so coordinates or distances in the 3D model to register must be known. The PnA problem, however, works on a slightly different basis because the only knowledge required for the model is a set of angles (scalar values). As a consequence, no positional data appear and the PnA problem can be seen as purely directional.

On the other hand, human-made environments not only are abundant in lines (fact already exploited in some works, e.g. for Structure from Motion (SfM) [8,

9]), but quite often these lines also present structural constraints such as parallelism or orthogonality. There are numerous works which exploit the structural characteristics of this kind of scenes (referred to as Manhattan world), specially in urban environments where clear dominant orthogonal directions are usually available and vanishing points can be robustly extracted [10, 11,12,13,14]. Other applications (like in [15]) use certain reduced configurations of parallel and orthogonal lines, which are particularly suitable for indoor environments where dominant directions are not always so readily available. Likewise, the solution to the P3oA problem can be exploited both in indoor and urban scenes since the orthogonality assumption of the problem holds also for human-made environments.

In this paper, we make two main contributions:

- First, we propose an algebraic general solution to the P3oA problem for a calibrated camera. Unlike previous proposals, this solution applies to any configuration of orthogonal 3D lines (even if they do not intersect) as depicted in Figure 1.
- Secondly, upon this P3oA solution we establish a new approach for the direct computation of the camera relative orientation. We apply RANSAC to find all the triplets of orthogonal lines and use them to compute a single rotation in closed form.

Therefore, the presented method allows for the estimation of the camera rotation, which is, per se, an essential step in a variety of applications. Methods which focus on the estimation of the rotation of the camera only are usually exploited as *visual gyroscopes*, and are useful for many tasks including humanoid stabilization [16] and ground/aerial vehicle control [17], among others. Furthermore, it is common to decouple the estimation of relative motion into two separates problems for the independent computation of rotation and translation. Therefore, our proposal can be also exploited as a partial solution to visual odometry, that may be complemented by any method for estimating the translational part.

2 Related work

Our first goal in this work is to develop an efficient solution to the P3oA problem. In a comprehensive analysis of how human vision naturally tends to perceive and interpret some patterns of lines as orthogonal in space, Barnard [18] points out the interest in somehow emulating this natural approach in automated vision. The solution he provides can tackle any triplet of orthogonal lines, its main limitation being its iterative nature.

Kanatani [19] finds a closed form solution for orthogonal three-line junctions prior transformation of the image to a canonical position, so that the common vertex is displaced to the camera principal axis. In [20], Wu *et al.* expand Kanatani's approach to the more general case of three lines meeting at arbitrary angles. Nevertheless, none of these closed-form solutions addresses the case of non intersecting lines. Furthermore, the fact that directions are parameterized with angular values may cause numerical instability as a consequence of non-linearities in trigonometric functions.

To the best of our knowledge, the only closed-form minimal solution able to solve the general P3oA problem is proposed by Mirzaei and Roumeliotis [13]. However, the operations involved in their minimal solution (e.g. computing the eigenvalue decomposition of a 8×8 matrix) make their approach more computationally expensive than necessary for this problem.

The P3oA problem could be also solved through an indirect approach by transforming it into an equivalent P3L problem, for which thorough studies have been reported [6,21]. One major limitation of this approach is that, unlike for P3L problems, P3oA lacks of any positional data. As a result only those P3L solutions which completely decouple the orientation computation from the positional data are usable. This fact is addressed in [20] where further analysis is done on the nature of different inverse perspective problems (PnP, PnL and PnA) and on which restatements are possible. In this sense, any P3L problem can be converted to P3A by computing the angles defined by the lines, but the opposite conversion is not always possible. Another drawback in coping with P3oA as a general P3L problem is that the more general the solution gets, the more complex it becomes too. Consequently, lighter and more efficient algorithms can be achieved for the P3oA problem if its inherent characteristics are fully exploited rather than considering it as a special case of a more complex problem.

On a different basis, much research has been done towards exploiting structural information of the scene. As a general rule, methods exploiting the assumption of structured world face a chicken-and-egg problem: If the searched configurations are known, the interest variable (e.g. vanishing points or camera rotation) can be computed. Reciprocally, if the interest variable is known, checking configurations can be readily done. A great deal of previous research in this area [22,23,24, 10,25,12] has focused mainly on the computation of Vanishing Points (VP) in the scene. Once the VPs are known, the camera orientation or other variables of interest are computed. Furthermore, some recent works [13,14] force the Manhattan world assumption during

the computation of the VPs, increasing the efficiency and precision of the estimation. We denominate the group of methods following this kind of approach *Vanishing Point*-based (VP-based) methods. A general limitation of these methods is that they strongly rely on the detection of Vanishing Points, thereby they fail when these are not found in the images.

A recent alternative which does not rely on the prior classification of all the lines in the scene into a set of Vanishing Points is proposed by Elqursh and Elgammal [15]. They use a primitive configuration, consisting of two parallel lines and a third line orthogonal to them, to compute the relative rotation between images. This configuration originates one of the two minimal problems encountered when solving the orientation in a Manhattan world from line observations [13], and a closed form solution for this is proposed (intrinsically equivalent to that appearing in [26]). Finally, all the valid primitive configurations in the image are found in a hypothesize-and-test framework and used to produce a more precise estimate of the relative rotation.

Our proposal for the computation of relative orientation is similar to Elqursh-Elgammal's, but a primitive configuration formed by three orthogonal lines is exploited instead. Using this configuration originates a P3oA problem, which is the other minimal problem encountered when solving orientation in a Manhattan world from line observations [13]. Because of this characteristic, we refer to our approach and Elqursh-Elgammal's as *Minimal Solution*-based methods.

3 Solution to the P3oA problem

In this section we will first formally state the P3oA problem and define some preliminary concepts and tools. Then, the particular case of P3oA with lines intersecting in a single point is solved under a novel approach (depicted in Algorithm 1). Afterwards, a more general solution for the case of non-intersecting orthogonal lines is presented.

3.1 Problem statement

Given the angles θ_{ij} formed by any pair of 3D lines (L_i, L_j) from a triplet $\{L_1, L_2, L_3\}$ and the image projection $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3\}$ of this triplet, the general P3A problem (non orthogonal angles) is that of finding the 3D direction $\boldsymbol{\nu}_k$ corresponding to each line, expressed in the camera reference frame (see Figure 1).

The vector $\boldsymbol{\nu}_k$ standing for each line direction is forced to lie on the unit 2-sphere S^2, so that $\|\boldsymbol{\nu}_k\| = 1$. With this parameterization the angular constraint can be written

$$\boldsymbol{\nu}_i^\top \cdot \boldsymbol{\nu}_j = \cos(\theta_{ij}) \tag{1}$$

where the possible pairs are $(i, j) = \{(1, 2), (2, 3), (3, 1)\}$.

The P3oA problem is then defined as the special case in which all 3D lines are orthogonal, so that (1) reduces to

$$\boldsymbol{\nu}_i^\top \cdot \boldsymbol{\nu}_j = 0 \tag{2}$$

3.2 Lines and interpretation planes

Let us consider an image line $\mathbf{l} \in \mathbb{P}^2$, characterized by a 3-vector in the projective space $\mathbb{P}^2$. This line corresponds to the projection of a 3D line L. The perspective projection model constrains the line L to lie in a particular plane passing through the origin of the camera's coordinate system and containing the image line $\mathbf{l}$ (see Figure 2). This plane Π is called the *interpretation plane* of line L. Since this plane contains the origin it can be fully characterized by its normal direction $\boldsymbol{n}$. For

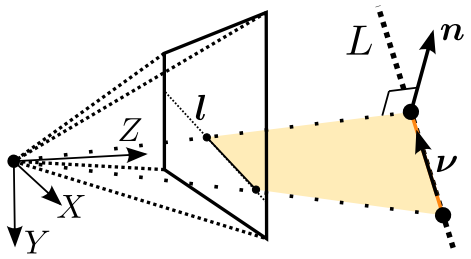


Fig. 2 The *interpretation plane* of a line L is defined as the plane which contains the camera origin and the line image $\mathbf{l}$

a projective camera with intrinsic calibration matrix $\boldsymbol{K}$ this normal is computed as [27]

$$\boldsymbol{n} = \frac{\boldsymbol{K}^\top \mathbf{l}}{\left\| \boldsymbol{K}^\top \mathbf{l} \right\|},$$

where normalization is applied to assure that the normal vector is unitary ($\boldsymbol{n} \in S^2$).

3.3 Parameterization of line directions

The problem unknowns, $\boldsymbol{\nu}_k \in S^2$, have 2 degrees of freedom each, summing up 6 unknowns.

Since each direction $\boldsymbol{\nu}_k$ is constrained to lie on the interpretation plane Π_k, a parameterization of $\boldsymbol{\nu}_k$ via a basis $\boldsymbol{\Omega}_k$ for Π_k is possible, that is,

$$\underset{3\times1}{\boldsymbol{\nu}_k} = \underset{3\times2}{\boldsymbol{\Omega}_k}\ \underset{2\times1}{\boldsymbol{\rho}_k}, \quad \boldsymbol{\Omega}_k = \text{Null}\left(\boldsymbol{n}_k^\top\right). \tag{3}$$

The square of the norm of the expression above is

$$\|\boldsymbol{\nu}_k\|^2 = \boldsymbol{\nu}_k^\top \boldsymbol{\nu}_k = \boldsymbol{\rho}_k^\top (\boldsymbol{\Omega}_k^\top \boldsymbol{\Omega}_k) \boldsymbol{\rho}_k,$$

so the normality condition $\|\boldsymbol{\nu}_k\| = 1$ is automatically fulfilled if the basis $\boldsymbol{\Omega}_k$ is taken orthonormal ($\boldsymbol{\Omega}_k^\top \boldsymbol{\Omega}_k = \mathbf{I}_2$) and $\boldsymbol{\rho}_k$ is constrained to lie in the unit 1-sphere S^1 ($\boldsymbol{\rho}_k^\top \boldsymbol{\rho}_k = 1$).

Thus, after applying the information encoded by each interpretation plane only 1 unknown is left for each 3D direction $\boldsymbol{\nu}_k$. The three remaning unknowns can be finally solved by imposing the angular constraints (2) of the P3oA problem, which applying the new parameterization in (3) reads

$$\boldsymbol{\rho}_i^\top \left(\boldsymbol{\Omega}_i^\top \boldsymbol{\Omega}_j \right) \boldsymbol{\rho}_j = 0 \tag{4}$$

where each matrix $\boldsymbol{\Omega}_i^\top \boldsymbol{\Omega}_j$ can be considered to stand for a non-symmetric bilinear map

$$\boldsymbol{B}_{ij} = \boldsymbol{\Omega}_i^\top \boldsymbol{\Omega}_j. \tag{5}$$

3.4 Revisiting the case of orthogonal meeting lines.

Firstly, we revisit the resolution of the special case in which all three image lines meet in a single point. In such cases the interpretation planes form a *pencil of* three *planes* (see Figure 3).

The *pencil direction* $\boldsymbol{t}$ is defined as the normalized direction of the pencil axis, which can be computed from the intersection of any two planes in the pencil

$$\boldsymbol{t} = \frac{\boldsymbol{n}_i \times \boldsymbol{n}_j}{\|\boldsymbol{n}_i \times \boldsymbol{n}_j\|} \tag{6}$$

A feasible basis for each interpretation plane Π_k in the pencil is

$$\boldsymbol{\Omega}_k = \begin{bmatrix} \boldsymbol{t} & \boldsymbol{t} \times \boldsymbol{n}_k \end{bmatrix} \tag{7}$$

This special basis will be referred to as the *pencil basis* for the intepretation plane Π_k. It can be seen from the definition of $\boldsymbol{t}$ in (6) that $\boldsymbol{t} \perp \boldsymbol{n}_k$, so $\|\boldsymbol{t} \times \boldsymbol{n}_k\| = 1$ and the pencil basis is *orthonormal*.

Under the parameterization of *pencil bases*, the bilinear forms in (5) becomes diagonal in an elegant and seamless way (see Appendix A):

$$\boldsymbol{B}_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & -\alpha_{ij} \end{bmatrix} \tag{8}$$

with

$$\alpha_{ij} = -\left(\boldsymbol{t} \times \boldsymbol{n}_i\right)^\top \left(\boldsymbol{t} \times \boldsymbol{n}_j\right) = -\boldsymbol{n}_i^\top \boldsymbol{n}_j \tag{9}$$

As a consequence, using pencil bases permits us to express each bilinear form $\boldsymbol{B}_{ij}$ through a single scalar value which equals, up to sign, the cosine of the angle between the interpretation planes Π_i and Π_j. The inherent duality when computing the angle between two planes disappears when a fixed representation $\boldsymbol{n}_k$ is taken for every plane normal.

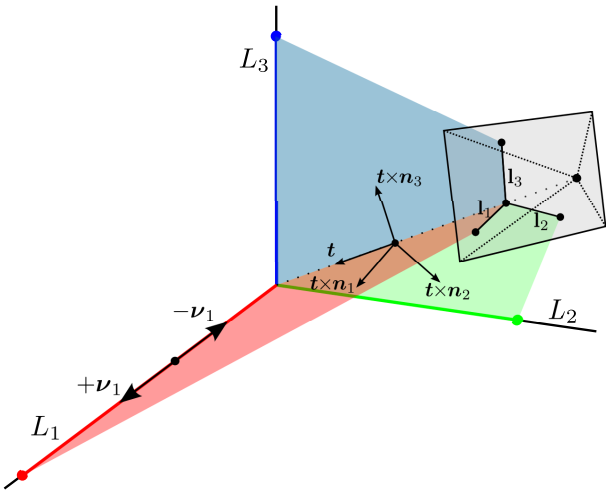


Fig. 3 P3oA problem for meeting lines: The intepretation planes form a pencil of planes. Both $+\boldsymbol{\nu}_1$ and $-\boldsymbol{\nu}_1$ represent the same direction

Under the pencil parameterization the condition (4) can be written as

$$\boldsymbol{\rho}_{i,1}\, \boldsymbol{\rho}_{j,1} - \alpha_{ij}\, \boldsymbol{\rho}_{i,2}\, \boldsymbol{\rho}_{j,2} = 0 \tag{10}$$

with $\boldsymbol{\rho}_{k,\cdot}$ the components of the vector $\boldsymbol{\rho}_k$:

$$\boldsymbol{\rho}_k = \begin{bmatrix} \boldsymbol{\rho}_{k,1} \\ \boldsymbol{\rho}_{k,2} \end{bmatrix}$$

Assuming that the second component of any $\boldsymbol{\rho}_k$ vector is non zero (which is always true for non degenerate cases, as discussed in Section 3.4.2), let us define the new unknowns

$$r_k = \frac{\boldsymbol{\rho}_{k,1}}{\boldsymbol{\rho}_{k,2}} \tag{11}$$

The problem in (10) can be rewritten then as a simpler system of quadratic equations

$$\begin{aligned} r_1 r_2 &= \alpha_{12} \\ r_2 r_3 &= \alpha_{23} \\ r_3 r_1 &= \alpha_{31} \end{aligned} \tag{12}$$

and taking into account the product of the three equations

$$(r_1 r_2 r_3)^2 = \alpha_{12}\alpha_{23}\alpha_{31} \Rightarrow r_1 r_2 r_3 = \pm\sqrt{\alpha_{12}\alpha_{23}\alpha_{31}}$$

the unknowns can be cleared as

$$r_k = \frac{\bar{\alpha}}{\alpha_{ij}}$$

with $\bar{\alpha} = \pm\sqrt{\alpha_{12}\alpha_{23}\alpha_{31}}$.

Therefore, each vector $\boldsymbol{\rho}_k$ can be solved from the calculated r_k by applying the condition $\|\boldsymbol{\rho}_k\| = 1$

$$\frac{\boldsymbol{\rho}_{k,1}}{\boldsymbol{\rho}_{k,2}} = \frac{\bar{\alpha}}{\alpha_{ij}}$$
$$\boldsymbol{\rho}_{k,1}^2 + \boldsymbol{\rho}_{k,2}^2 = 1$$

and the reduced solution to the P3oA problem takes a closed form expression

$$\boldsymbol{\rho}_k = \pm\frac{1}{\sqrt{\bar{\alpha}^2 + \alpha_{ij}^2}}\begin{bmatrix}\bar{\alpha}\\ \alpha_{ij}\end{bmatrix} \tag{13}$$

Once the reduced vector $\boldsymbol{\rho}_k$ is known, the corresponding 3D direction $\boldsymbol{\nu}_k$ can be recovered from (3) as

$$\boldsymbol{\nu}_k = \begin{bmatrix}\boldsymbol{t} & \boldsymbol{t}\times\boldsymbol{n}_k\end{bmatrix}\boldsymbol{\rho}_k.$$

Hence, the resolution for the case of intersecting lines reduces to the steps given in Algorithm 1.

Algorithm 1 P3oA solution for intersecting lines

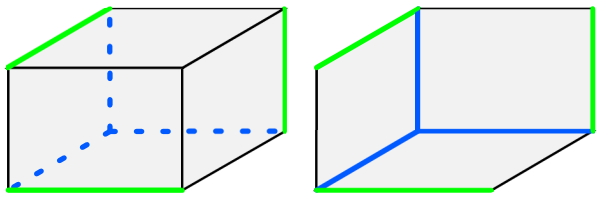
Data:
$\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3$ ▷ interpr. planes normals
s ▷ Necker's sign
Require: $\det(\begin{bmatrix}\boldsymbol{n}_1 & \boldsymbol{n}_2 & \boldsymbol{n}_3\end{bmatrix}) = 0$

function P3oA_MEETING($\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3, s$)
  $\alpha_{12} \leftarrow -\boldsymbol{n}_1^\top\boldsymbol{n}_2$, $\alpha_{23} \leftarrow -\boldsymbol{n}_2^\top\boldsymbol{n}_3$, $\alpha_{31} \leftarrow -\boldsymbol{n}_3^\top\boldsymbol{n}_1$
  $\bar{\alpha} \leftarrow s\cdot\sqrt{\alpha_{12}\alpha_{23}\alpha_{31}}$
  for $k \leftarrow 1,3$ **do**
    $\boldsymbol{\rho}_k \leftarrow \frac{1}{\sqrt{\bar{\alpha}^2+\alpha_{ij}^2}}\begin{bmatrix}\bar{\alpha}, & \alpha_{ij}\end{bmatrix}^\top$ ▷ **get** reduced solution
    $\boldsymbol{\Omega}_k \leftarrow \begin{bmatrix}\boldsymbol{t} & \boldsymbol{t}\times\boldsymbol{n}_k\end{bmatrix}$ ▷ **build** k-th pencil basis
    $\boldsymbol{\nu}_k \leftarrow \boldsymbol{\Omega}_k\,\boldsymbol{\rho}_k$ ▷ **undo** pencil parameterization
  end for
end function

3.4.1 Analysis of multiple solutions

The expression (13) leads to a total of 16 solutions due to 4 sign ambiguities. However, the 3 outer signs affecting each $\boldsymbol{\rho}_k$ vector are due to the inherent ambiguity in line direction since, given a point, the lines with directions $+\boldsymbol{\nu}_k$ and $-\boldsymbol{\nu}_k$ represent exactly the same entity (see Figure 3).

Therefore, only the sign of the parameter $\bar{\alpha}$ requires our attention. This sign is linked to the existence of two distinct sets of directions which are solution to the same P3oA problem. This duality, which has been known for a long time, is named the *Necker's cube illusion* and has been already pointed out by other authors when solving the P3A problem [20]. This illusion is displayed in Figure 4, where we can notice that exactly the same three segments may be perceived as a concave or a convex trihedron of two different cubes.



(a) Natural perception (b) Unusual perception

Fig. 4 Necker's cube illusion: For the same image two interpretations are possible for both meeting (blue) and non-meeting (green) cases

3.4.2 Analysis of degenerate cases

A P3oA problem is degenerate if two lines have the same projection in the image. In such a case, the two interpretation planes corresponding to the coincident lines become the same, and the remaining interpretation plane is orthogonal to the other two (see Appendix B for a complete proof). Note that the orthogonality between interpretation planes does not involve that lines in the image are orthogonal to each other, since

$$\boldsymbol{n}_i^\top\boldsymbol{n}_j \propto \mathbf{l}_i^\top(\boldsymbol{K}\boldsymbol{K}^\top)\mathbf{l}_j = 0 \nRightarrow \mathbf{l}_i^\top\mathbf{l}_j = 0.$$

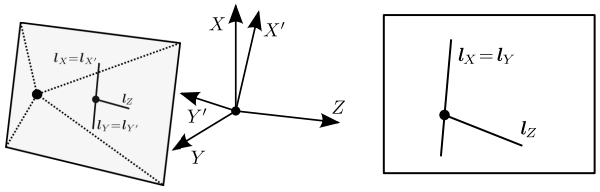


Fig. 5 Degenerate case: $\mathbf{l}_X$ and $\mathbf{l}_Y$ are degenerate, so the relation $\boldsymbol{n}_i \parallel \boldsymbol{n}_j \perp \boldsymbol{n}_k$ stands. Note that the orthogonality condition is between interpretation planes, not image lines. There exist infinite other pairs of lines X' and Y' contained in the plane XY which would produce the same image

Let us assume, without loss of generality, that the lines L_i and L_j are the source of the degeneracy (see Figure 5 above). Then, according to the results in Appendix B, the interpretation planes' normals fulfill

$$\boldsymbol{n}_i \parallel \boldsymbol{n}_j,\ \boldsymbol{n}_k \perp \boldsymbol{n}_i,\ \boldsymbol{n}_k \perp \boldsymbol{n}_j$$

and according to (9) the pencil basis parameters become

$$\alpha_{ij} = 1$$
$$\alpha_{jk} = \alpha_{ki} = 0$$

For this particular configuration, the system of quadratic equations (12) reads

$$r_i r_j = 1$$
$$r_j r_k = 0$$
$$r_k r_i = 0$$

from which it is easy to conclude that $r_k = 0$. As a result

$$\left.\begin{array}{l} r_k = \boldsymbol{\rho}_{k,1}/\boldsymbol{\rho}_{k,2} = 0 \\ \|\boldsymbol{\rho}_k\|^2 = \boldsymbol{\rho}_{k,1}^2 + \boldsymbol{\rho}_{k,2}^2 = 1 \end{array}\right\} \Rightarrow \boldsymbol{\rho}_k = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

This allows us to fully recover the direction of L_k from (3) as

$$\boldsymbol{\nu}_k = \boldsymbol{t} \times \boldsymbol{n}_k.$$

The pencil direction $\boldsymbol{t}$ can be computed as

$$\boldsymbol{t} = \frac{\boldsymbol{n}_k \times \boldsymbol{n}_i}{\|\boldsymbol{n}_k \times \boldsymbol{n}_i\|},$$

and, since $\boldsymbol{n}_k \perp \boldsymbol{n}_i$, we have $\|\boldsymbol{n}_k \times \boldsymbol{n}_i\| = 1$ and the final result is

$$\begin{aligned} \boldsymbol{\nu}_k &= (\boldsymbol{n}_k \times \boldsymbol{n}_i) \times \boldsymbol{n}_k \\ &= -(\boldsymbol{n}_i^\top \boldsymbol{n}_k)\, \boldsymbol{n}_k + (\boldsymbol{n}_k^\top \boldsymbol{n}_k)\, \boldsymbol{n}_i = \boldsymbol{n}_i. \end{aligned}$$

However, the unknowns r_i and r_j cannot be computed with the remaining data. Instead, only the constraint that the corresponding directions $\boldsymbol{\nu}_i$ and $\boldsymbol{\nu}_j$ are orthogonal is kept. As a result, the direction corresponding to the non-degenerate line L_k can be fully recovered, whereas those of the degenerate lines L_i and L_j are only constrained to be a pair of orthogonal directions inside the degenerate interpretation plane:

$$\begin{aligned} &\boldsymbol{\nu}_k = \boldsymbol{n}_i = \boldsymbol{n}_j \\ &\boldsymbol{\nu}_i \perp \boldsymbol{\nu}_j, \quad \left[\boldsymbol{\nu}_i, \boldsymbol{\nu}_j\right] = \text{Null}\left(\boldsymbol{\nu}_k^\top\right) \in \Pi_i = \Pi_j \end{aligned}$$

A second special degenerate configuration would happen when the assumption $\boldsymbol{\rho}_{k,2} \neq 0$, made when defining r_k in (11), does not hold. This however would mean that for the line L_k the direction is

$$\boldsymbol{\nu}_k = \begin{bmatrix} \boldsymbol{t} & \boldsymbol{t} \times \boldsymbol{n}_k \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \boldsymbol{t} \tag{14}$$

but the pencil direction $\boldsymbol{t}$ and the line direction $\boldsymbol{\nu}_k$ can only be the same if the camera center lies in L_k and, in such a case, the line projects into the image as a point. This case lacks interest for us since we expect lines to be segments in the image, not points.

3.5 Extension to the case of non-meeting orthogonal lines

The solution to the P3oA problem with three meeting lines can be taken as the starting point to solve the P3oA general case, that is, when the lines in image do not intersect at a single point.

Let us assume that the image projections $\mathbf{l}_k$ of three generic orthogonal 3D lines $\{L_1, L_2, L_3\}$ are given. Their corresponding interpretation planes $\{\Pi_1, \Pi_2, \Pi_3\}$ no longer intersect at a single line, so there exists no set of bases $\boldsymbol{\Omega}_k$ for which three diagonal bilinear forms, as those of (8), arise simultaneously. It is possible to build, however, an equivalent problem replacing L_k by an auxiliary line L_* with its same direction, whose image projection intersects with those of L_i and L_j at a single point (see Figure 6). That is equivalent to impose that the interpretation planes form a pencil of planes, which in terms of the plane normals is fulfilled when

$$\det(\left[\boldsymbol{n}_i\, \boldsymbol{n}_j\, \boldsymbol{n}_*\right]) = 0. \tag{15}$$

From now on, this equivalent problem will be referred

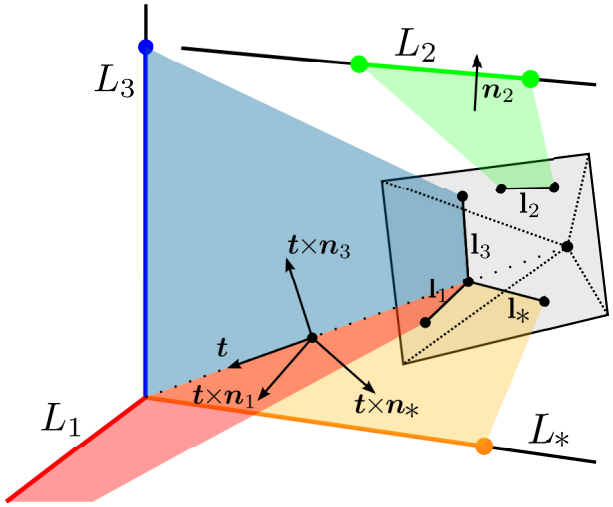


Fig. 6 General P3oA problem: It is always possible to create a equivalent virtual problem with lines meeting at the same point. Here $\mathbf{l}_*$ replaces $\mathbf{l}_2$

to as the *virtual* problem, and the auxiliary line L_* and all related variables will be also tagged as *virtual*, with the symbol $*$. The solution to the virtual problem would be the one already presented for the special case of meeting lines (13).

Although the image $\mathbf{l}_*$ for the virtual line is not known, by definition it must fulfill the meeting condition in (15), so the corresponding normal $\boldsymbol{n}_*$ can be expressed as a linear combination of $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$, or equivalently, $\boldsymbol{n}_*$ must be orthogonal to the pencil direction $\boldsymbol{t}$, so it can be expressed as

$$\boldsymbol{n}_* = \boldsymbol{\Omega}_{\boldsymbol{t}}\, \boldsymbol{\delta} \tag{16}$$

with $\boldsymbol{\Omega}_{\boldsymbol{t}}$ being an orthonormal basis for $\boldsymbol{t}^\top$ right nullspace and $\boldsymbol{\delta}$ a reduced representation for $\boldsymbol{n}_*$, so that $\boldsymbol{\delta} \in S^1$.

Using the parameterization given in (3)

$$\boldsymbol{\nu}_k = \boldsymbol{\nu}_* = \boldsymbol{\Omega}_*\, \boldsymbol{\rho}_*$$

and substituting the expressions for the pencil basis (7) and the meeting-case solution (13), the direction for line L_* (and so for L_k) is

$$\boldsymbol{\nu}_k = \boldsymbol{\nu}_* = \frac{\pm 1}{\sqrt{\bar{\alpha}^2 + \alpha_{ij}^2}} \underbrace{\begin{bmatrix} \boldsymbol{t} & [\boldsymbol{t}]_\times \boldsymbol{n}_* \end{bmatrix}}_{3\times 2} \underbrace{\begin{bmatrix} \bar{\alpha} \\ \alpha_{ij} \end{bmatrix}}_{2\times 1}, \tag{17}$$

where all the parameters are relative to the virtual problem (so they depend on the unknown $\boldsymbol{\delta}$ too). Since we do know the interpretation plane for the original L_k, the extra condition

$$\boldsymbol{n}_k^\top \, \boldsymbol{\nu}_k = 0$$

can be applied to solve $\boldsymbol{\delta}$. Replacing $\boldsymbol{\nu}_k$ with (17) in the condition above, the equation to solve reads

$$\bar{\alpha} \cdot \boldsymbol{n}_k^\top \boldsymbol{t} + \alpha_{ij} \cdot \boldsymbol{n}_k^\top \, [\boldsymbol{t}]_\times \boldsymbol{n}_* = 0 \tag{18}$$

with

$$\begin{aligned} &\bar{\alpha} = \pm\sqrt{\alpha_{*i}\alpha_{ij}\alpha_{j*}} \\ &\alpha_{*i} = -\boldsymbol{n}_*^\top \boldsymbol{n}_i, \; \alpha_{ij} = -\boldsymbol{n}_i^\top \boldsymbol{n}_j, \; \alpha_{j*} = -\boldsymbol{n}_j^\top \boldsymbol{n}_* \end{aligned} \tag{19}$$

The two solutions of equation (18) are contained in the set of 4 solutions of an equivalent quadratic equation

$$\boldsymbol{n}_*^\top \boldsymbol{Q} \, \boldsymbol{n}_* = 0 \tag{20}$$

with

$$\boldsymbol{Q} = \frac{1}{2} \frac{(\boldsymbol{n}_k^\top \boldsymbol{t})^2}{\alpha_{ij}} \left(\boldsymbol{n}_i \boldsymbol{n}_j^\top + \boldsymbol{n}_j \boldsymbol{n}_i^\top \right) + [\boldsymbol{t}]_\times \left(\boldsymbol{n}_k \boldsymbol{n}_k^\top \right) [\boldsymbol{t}]_\times \tag{21}$$

as deducted in Appendix C.

We substitute in the equivalent equation (20) the parameterization given in (16) so that the final problem to solve becomes

$$\boldsymbol{\delta}^\top \boldsymbol{Q}_{\boldsymbol{t}} \, \boldsymbol{\delta} = 0 \tag{22}$$

defined by the *equivalent quadratic*

$$\boldsymbol{Q}_{\boldsymbol{t}} = \boldsymbol{\Omega}_{\boldsymbol{t}}^\top \boldsymbol{Q} \, \boldsymbol{\Omega}_{\boldsymbol{t}} \tag{23}$$

The symmetric matrix above can be diagonalized, e.g. via SVD or eigendecomposition, into

$$\boldsymbol{Q}_{\boldsymbol{t}} = \boldsymbol{U} \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \boldsymbol{U}^\top \tag{24}$$

with $\boldsymbol{U}$ a 2×2 orthonormal basis. From this decomposition it can be concluded that the solutions to (22) fulfilling $\boldsymbol{\delta} \in S^1$ are

$$\boldsymbol{\delta} = \frac{1}{\sqrt{|d_1| + |d_2|}} \boldsymbol{U} \begin{bmatrix} +\sqrt{|d_2|} \\ \pm\sqrt{|d_1|} \end{bmatrix} \tag{25}$$

as long as $\text{sign}(d_1) \neq \text{sign}(d_2)$. As a result, it is a necessary condition that the 2×2 quadratic form $\boldsymbol{Q}_{\boldsymbol{t}}$ is non-definite for a solution to the P3oA problem to exist. The case in which $\boldsymbol{Q}_{\boldsymbol{t}}$ is semidefinite (non full-rank) corresponds to the previous case of meeting lines. Thus, from now on we consider the cases for which $\boldsymbol{Q}_{\boldsymbol{t}}$ is indefinite.

Once $\boldsymbol{\delta}$ is solved, the seeked virtual normal $\boldsymbol{n}_*$ can be recovered from (16). Two possible $\boldsymbol{n}_*$ are obtained due to the sign indeterminacy in (25), each corresponding to a mathematically feasible world model. Furthermore, each of them outcomes two possible solutions because of the duality in the P3oA problem. So, the number of reachable solutions adds up to 4, from which 2 are false solutions that appear when squaring the radical equation. However, for a given $\boldsymbol{n}_*$ solution, the value $|\bar{\alpha}|$ is completely defined by (19) and the correct sign for $\bar{\alpha} = s\,|\bar{\alpha}|$ can be extracted from the original radical equation (18)

$$s = -\,\text{sign}\left(\frac{\alpha_{ij} \cdot \boldsymbol{n}_k^\top \, [\boldsymbol{t}]_\times \, \boldsymbol{n}_*}{|\bar{\alpha}| \cdot \boldsymbol{n}_k^\top \boldsymbol{t}}\right) \tag{26}$$

Once both $\boldsymbol{n}_*$ and the correct $\text{sign}(\bar{\alpha})$ are known for each case, the simple solution depicted in Algorithm 1 can be applied to solve each virtual problem, thereby obtaining two possible solutions to the original problem.

The general P3oA procedure described above is depicted in Algorithm 2.

4 Application of the P3oA solution to visual odometry

In this section the P3oA problem is exploited to obtain a minimal solution for the camera relative rotation based on the observation of a single triplet of orthogonal lines in two different frames. We propose, then, a robust framework which employs all the orthogonal configurations available to estimate the camera relative (incremental) rotation. Therefore, the presented algorithm can be used as a *visual gyroscope*.

Furthermore, since the estimation of relative motion is often decoupled in estimation of rotation and translation, our solution can be used to address visual odometry if it is complemented by any method for estimating the translational part (e.g. the proposal in [15]).

4.1 Minimal solution for the relative rotation

Let $\{\boldsymbol{\nu}_k\}_{k=1}^3$ be a set of three orthogonal directions coming from the minimal solution of a P3oA problem (see

Algorithm 2 General P3oA algorithm

Data:
$\boldsymbol{K}$ ▷ camera intrisic calibration
$\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$ ▷ homogeneous image lines
Result:
$\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3$ ▷ 3D directions in camera frame

Initialization:
for $k \leftarrow 1,3$ **do**
  $\boldsymbol{n}_k \leftarrow \mathbb{N}_{\mathrm{S}}(\boldsymbol{K}^\top \mathbf{l}_k)$ ▷ get interpr. plane normal
end for
General P3oA resolution:
if $\det([\boldsymbol{n}_1\, \boldsymbol{n}_2\, \boldsymbol{n}_3]) = 0$ **then** ▷ if intersecting lines
  for $m \leftarrow 1,2$ **do**
    $\{\boldsymbol{\nu}_k^{(m)}\}_{k=1}^3 \leftarrow$ P3oA_meet$(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3, (-1)^m)$
  end for
else ▷ if non-intersecting lines
  take two normals (e.g. $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$)
  $\boldsymbol{t} \leftarrow \mathbb{N}_{\mathrm{S}}(\boldsymbol{n}_1 \times \boldsymbol{n}_2)$ ▷ *pencil direction*
  $\boldsymbol{\Omega_t} \leftarrow \mathrm{Null}\left(\boldsymbol{t}^\top\right)$ ▷ pencil nullspace
  build *equivalent quadratic* $\boldsymbol{Q_t}$ ▷ see (21) and (23)
  solve $\boldsymbol{\delta}^{(m)}$ from SVD$(\boldsymbol{Q_t})$ ▷ see (25)
  for $m \leftarrow 1,2$ **do**
    $\boldsymbol{n}_* \leftarrow \boldsymbol{\Omega_t}\, \boldsymbol{\delta}^{(m)}$ ▷ compute virtual normals
    compute s sign for $\boldsymbol{n}_*$ ▷ see (26)
    $\{\boldsymbol{\nu}_k^{(m)}\}_{k=1}^3 \leftarrow$ P3oA_meet$(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_*, s)$
  end for
end if
* $\mathbb{N}_{\mathrm{S}}(\cdot) \equiv \frac{\cdot}{\|\cdot\|}$ ▷ Abbreviation for normalization

Section 3). We stack them as the columns of an orthogonal *direction matrix* $\boldsymbol{V} \in \mathrm{O}(3)$. As shown in Section 3.4.1, there are 16 valid solutions for a minimal P3oA problem: Due to Necker's duality the solution produces two geometrically distinct sets of directions $\boldsymbol{V}^{(m)}$, where we use the upper index $m = \{1, 2\}$ to distinguish both Necker alternatives. On the other hand, each direction is recovered up to sign. The sign ambiguities can be gathered into a diagonal sign matrix, so that the whole set of 16 solutions for a minimal P3oA problem is represented by the single expression $\boldsymbol{V}^{(m)}\boldsymbol{S}$, where

$$\boldsymbol{S} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix}, \quad s_i = \pm 1. \tag{27}$$

Let $\boldsymbol{R}_{\mathrm{rel}}$ and $\boldsymbol{t}_{\mathrm{rel}}$ be the relative rotation and translation, respectively, of the camera in the second frame with respect to (wrt) the camera coordinate system in the first frame. The relative rotation between the frames can be solved from the directions of the triplet of orthogonal lines observed in both frames. The 3D directions coming from the solution of the P3oA problem in each frame are

$$\boldsymbol{V}_1^{(m_1)}\boldsymbol{S}_1,\ \boldsymbol{V}_2^{(m_2)}\boldsymbol{S}_2, \tag{28}$$

so one could think that the combination of both would produce *256* different *solutions* for $\boldsymbol{R}_{\mathrm{rel}}$. This is not so, though. All the possible solutions for $\boldsymbol{R}_{\mathrm{rel}}$ can be condensed into the expression

$$\boldsymbol{R}_{\mathrm{rel}} = \boldsymbol{V}_1^{(m_1)}\boldsymbol{S}_1\boldsymbol{S}_2^\top {\boldsymbol{V}_2^{(m_2)}}^\top, \tag{29}$$

and this simplified into

$$\boldsymbol{R}_{\mathrm{rel}} = \boldsymbol{V}_1^{(m_1)}\, \boldsymbol{S}\, {\boldsymbol{V}_2^{(m_2)}}^\top, \tag{30}$$

where

$$\boldsymbol{S} = \boldsymbol{S}_1\boldsymbol{S}_2^\top = \begin{bmatrix} s_{1,1}s_{2,1} & 0 & 0 \\ 0 & s_{1,2}s_{2,2} & 0 \\ 0 & 0 & s_{1,3}s_{2,3} \end{bmatrix} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix}.$$

We also impose the constraint $\det(\boldsymbol{R}_{\mathrm{rel}}) = +1$ to assure that $\boldsymbol{R}_{\mathrm{rel}}$ is a valid rotation:

$$\begin{aligned} \det(\boldsymbol{R}_{\mathrm{rel}}) &= \det(\boldsymbol{V}_1^{(m_1)}\, \boldsymbol{S}\, {\boldsymbol{V}_2^{(m_2)}}^\top) \\ &= \det(\boldsymbol{V}_1^{(m_1)}) \det(\boldsymbol{S}) \det(\boldsymbol{V}_2^{(m_2)}) \\ &= s_1 s_2 s_3 \det(\boldsymbol{V}_1^{(m_1)}) \det(\boldsymbol{V}_2^{(m_2)}) = +1. \end{aligned}$$

As a result, the condensed expression for all the possible solutions can be finally reduced to

$$\begin{aligned} &\boldsymbol{R}_{\mathrm{rel}} = \boldsymbol{V}_1^{(m_1)}\, \boldsymbol{S}\, {\boldsymbol{V}_2^{(m_2)}}^\top \\ &s_1, s_2 = \pm 1, \quad s_3 = s_1 s_2 \det(\boldsymbol{V}_1^{(m_1)}) \det(\boldsymbol{V}_2^{(m_2)}). \end{aligned} \tag{31}$$

The conclusion is that 4 degrees of freedom (2 for Necker duality and 2 for direction signs) are retained and there are $2^4 = 16$ solutions for the relative rotation. However, when working with keyframes in a video sequence as in visual odometry [28], it makes sense to assume that the relative rotation between any two frames is well below 90 degrees. We will show next how this *small rotation assumption* allows us to further reduce the solution multiplicity to only two potential solutions. If it were not the case that rotation angle was small, but big changes of view could occur, other approaches could be adopted such as examining the gradient at the segments to determine if they are flipped between the two images [15].

4.1.1 Sign ambiguity

Under the *small rotation assumption*, the sign matrix $\boldsymbol{S}$ in (31) is taken so that the rotation $\boldsymbol{R}_{\mathrm{rel}}$ is closest to identity, that is:

$$\boldsymbol{S} = \mathrm{diag}(\mathrm{sign}(\mathrm{diag}({\boldsymbol{V}_1^{(m_1)}}^\top \mathbf{I}_3 \boldsymbol{V}_2^{(m_2)}))).$$

As a result, the multiplicity coming from the sign ambiguities (two degrees of freedom) is removed and only 4 solutions remain due to Necker's duality.

4.1.2 Necker ambiguity

Let $\boldsymbol{V}_i$ and $\boldsymbol{V}_i^*$ stand for the true and dual (false) solution, respectively, for a certain instance of the P3oA problem due to Necker duality. From (31), the four possible combinations of Necker modalities are

$$(\boldsymbol{V}_1, \boldsymbol{V}_2), (\boldsymbol{V}_1^*, \boldsymbol{V}_2), (\boldsymbol{V}_1, \boldsymbol{V}_2^*), (\boldsymbol{V}_1^*, \boldsymbol{V}_2^*).$$

Under the *small rotation assumption*, half of the combinations can be ruled out through a simple heuristic rule: corresponding Necker configurations are closer than non corresponding ones. Different metrics can be applied to measure this distance, e.g. the angular distance defined in equation (34).

Once this distinction is applied, only the two combinations $\{(\boldsymbol{V}_1, \boldsymbol{V}_2), (\boldsymbol{V}_1^*, \boldsymbol{V}_2^*)\}$ remain. When more solutions are available, the selection between these two options can be performed within the RANSAC step, as described in the next section. However, if no parallax exists between frames both pairs provide exactly the same solution, as explained in Appendix D.

4.2 Robust estimation from multiple configurations

The robust estimation of the relative rotation using all the triplets of orthogonal lines available in the scene comprises a series of steps: Firstly, all possible triplets of segment matches among the images are considered, and this set is reduced by pruning non feasible combinations. The remaining triplets are classified in a RANSAC framework and finally a global estimation of the rotation is performed from all the inlier triplets.

4.2.1 Filtering feasible candidate triplets

In the proposed approach (Algorithm 3) it is first necessary to generate all possible 3-combinations from, say, N^* segment matches among images, whose number adds up to

$$\binom{N^*}{3} = \frac{N^*!}{3!(N^*-3)!} = \frac{1}{6}N^*(N^*-1)(N^*-2),$$

so the number of candidates grows with the cube of N^*. However, many of these generated combinations are not feasible candidates. Therefore, it is most interesting to prune the large list of candidates in advance, which can be done by checking the solvability of the arising numerical problem. This simple test allows us to rule out many of the generated candidate triplets, so that only N of the original N^* remain.

Depending on the case to be solved, a different condition stands for a given configuration of normals to be solvable. For meeting lines (Section 3.4), since the solution depends on the parameter

$$\bar{\alpha} = \sqrt{\alpha_{12}\alpha_{23}\alpha_{31}}$$

it is clear that the product inside the square root should be positive and the condition becomes

$$\alpha_{12}\alpha_{23}\alpha_{31} > 0 \tag{32}$$

As for non meeting lines (Section 3.5), the equivalent quadratic $\boldsymbol{Q_t}$ must be indefinite, which in the 2×2 case reduces to

$$\det(\boldsymbol{Q_t}) < 0 \tag{33}$$

Each triplet of matches comprises two triplets of normals, one per image, and the conditions above must be fulfilled by both.

4.2.2 Finding true configurations

From now on, $\boldsymbol{R}$ will stand for the relative rotation between the two camera frames. The direction matrices for the k-th triplet of lines in the first and second image are, respectively, $\boldsymbol{V}_{1,k}^{(m)}$ and $\boldsymbol{V}_{2,k}^{(m)}$, and the corresponding rotation candidate computed by the minimal solution (in Section 4.1) is denoted as $\boldsymbol{R}_k^{(m)}$.

From the pruned list of N triplets, $2N$ candidates $\boldsymbol{R}_k^{(m)}$ are computed for the relative rotation $\boldsymbol{R}$, many of which may be outliers generated due to non orthogonal combinations, false Necker configurations, wrong matchings, bad segment detections, etc. Thus, RANSAC is applied to find the relative rotation with maximum support among all the available candidates. The goodness of this approach bases on the sensible assumption that triplets which are not orthogonal will vote for random rotations and will not support a concrete solution. Since only one sample is necessary for each RANSAC step, the number of iterations and the computational cost is relatively low.

A metric is needed in the RANSAC framework to check if the hypothesis is fullfilled, for which the angular distance for rotations [29] has been chosen

$$\begin{aligned} \mathrm{d}(\boldsymbol{R}_a, \boldsymbol{R}_b) &\equiv \mathrm{d}_\angle(\boldsymbol{R}_a, \boldsymbol{R}_b) = \theta \\ &= 2\arcsin\left(\frac{\|\boldsymbol{R}_a - \boldsymbol{R}_b\|_{\mathfrak{F}}}{2\sqrt{2}}\right). \end{aligned} \tag{34}$$

Here $\|\cdot\|_{\mathfrak{F}}$ stands for the *Frobenius norm* defined as

$$\|\boldsymbol{A}\|_{\mathfrak{F}} = \sqrt{\mathrm{trace}(\boldsymbol{A}^\top \boldsymbol{A})}. \tag{35}$$

The necessary threshold $\tilde{\theta}$ to decide if two rotations stand for the same solution is set to $\tilde{\theta} \approx 1.5^\circ$, since the variance of the solutions due to noise tends to be of that order.

Algorithm 3 Incremental rotation computation

Data:
$\boldsymbol{K}$ ▷ camera intrisic calibration
$\{\mathrm{l}_{1,k}\}$ ▷ lines in image #1
$\{\mathrm{l}_{2,k}\}$ ▷ lines in image #2
$\boldsymbol{M}$ ▷ $2 \times N^*$ matches matrix
Result:
$\boldsymbol{R}$ ▷ Rotation of the camera between frames

Compute rotation candidates:
generate all triplets of matches $\{\boldsymbol{\Gamma}\}_{k=1}^{N^*} \leftarrow \text{nchoosek}(\boldsymbol{M}, 3)$
filter non-solvable triplets: ▷ see Section 4.2.1 $\{\boldsymbol{\Gamma}\}_{k=1}^{N} \leftarrow \{\boldsymbol{\Gamma}\}_{k=1}^{N^*}$
for $k \leftarrow 1, N$ **do** ▷ for each *triplet*
  for $i \leftarrow 1, 2$ **do** ▷ for each *image*
    $\boldsymbol{V}_i \leftarrow \text{P3oA}(\mathrm{l}_1, \mathrm{l}_2, \mathrm{l}_3)$ ▷ see Algorithm 2
  end for
  $\boldsymbol{R}_k \leftarrow \boldsymbol{V}_1 \, \boldsymbol{S} \, \boldsymbol{V}_2^\top$ ▷ see Section 4.1
end for
RANSAC filtering:
find candidate with maximum support and inliers $\boldsymbol{I}$: $\boldsymbol{I} \leftarrow \text{argmax}(\#\{\boldsymbol{R} : \mathrm{d}(\boldsymbol{R}_k, \boldsymbol{R}) < \theta, \, \boldsymbol{R} \in \{\boldsymbol{R}_k\}\})$
Global optimization:
$\Upsilon_1 \leftarrow \left[\boldsymbol{V}_{1,\boldsymbol{I}}\right], \Upsilon_2 \leftarrow \left[\boldsymbol{V}_{2,\boldsymbol{I}}\right]$ ▷ stack 3D directions
$\boldsymbol{U}_{\boldsymbol{R}} \, \boldsymbol{\Sigma}_{\boldsymbol{R}} \, \boldsymbol{V}_{\boldsymbol{R}} = \Upsilon_1 \Upsilon_2^\top$ ▷ compute SVD
$\boldsymbol{R} = \boldsymbol{U}_{\boldsymbol{R}} \boldsymbol{V}_{\boldsymbol{R}}^\top$ ▷ Procrustes resolution

4.2.3 Global refinement

To increase the precision of the method, the final estimate of the camera rotation $\boldsymbol{R}$ is computed through the optimization of a cost function which involves all the valid candidates obtained from the prior step.

Formally, we address the task as a least squares optimization problem in which we minimize the sum of the squared distances between corresponding direction matrices after transformation by $\boldsymbol{R}$, that is

$$\hat{\boldsymbol{R}} = \underset{\boldsymbol{R}}{\text{argmin}}\Big(\sum_{k \in \text{inliers}} \text{dist}^2(\boldsymbol{V}_{1,k}, \boldsymbol{R}\,\boldsymbol{V}_{2,k}) \Big). \tag{36}$$

When working with rotations, there are several common alternatives to measure distances [29]. The *angular distance* defined in (34) is usually considered the most *natural* metric for rotations in SO(3). However, we prefer here the *chordal distance* [29] defined as

$$\mathrm{d}_{chord}(\boldsymbol{R}_a, \boldsymbol{R}_b) = \|\boldsymbol{R}_a - \boldsymbol{R}_b\|_{\mathfrak{F}} \tag{37}$$

because due to its quadratic nature it yields simpler expressions than the angular distance. For the same reasons, this metric has been used before in [30]. Interestingly, when the residuals are small (as one would expect in the global optimum) both metrics are equivalent up to a first-order approximation [29,30]:

$$\mathrm{d}_{chord}^2 = 8 \sin^2(\theta/2) \approx 8(\theta/2)^2 = 2\,\mathrm{d}_{\angle}{}^2 \tag{38}$$

The substitution of the chosen metric (37) into the least squares problem (36) yields the optimization problem

$$\hat{R} = \underset{\boldsymbol{R}}{\text{argmin}}\Big(\sum_{k \in \text{inliers}} \|\boldsymbol{V}_{1,k} - \boldsymbol{R}\,\boldsymbol{V}_{2,k}\|_{\mathfrak{F}}^2 \Big) \tag{39}$$

This is equivalent to the *orthogonal Procrustes problem* [31] defined as

$$\hat{\boldsymbol{R}} = \underset{\boldsymbol{R}}{\text{argmin}}(\|\Upsilon_1 - \boldsymbol{R}\,\Upsilon_2\|_{\mathfrak{F}}^2), \tag{40}$$

where Υ_1 and Υ_2 are the concatenation of the inlier direction matrices $\boldsymbol{V}_{1,k}$ and $\boldsymbol{V}_{2,k}$, respectively. This problem admits a closed-form solution [32] and the global minimum is attained for $\boldsymbol{R} = \boldsymbol{U}_{\boldsymbol{R}} \boldsymbol{V}_{\boldsymbol{R}}^\top$, where $\boldsymbol{U}_{\boldsymbol{R}} \, \mathbf{S}_{\boldsymbol{R}} \boldsymbol{V}_{\boldsymbol{R}}^\top$ is the Singular Value Decomposition of $\Upsilon_1 \Upsilon_2^\top$.

5 Experimental Results

In this section we evaluate and analyze the performance of our two main contributions: A minimal solution to the P3oA problem and a visual gyroscope based on the use of this minimal solution. Thus, the experiments are divided into two main groups. First we focus on the comparison of the proposed minimal solution with other existing alternatives, and then we present an extensive set of experiments in which its application to a visual gyroscope is tested and compared to state-of-the-art methods.

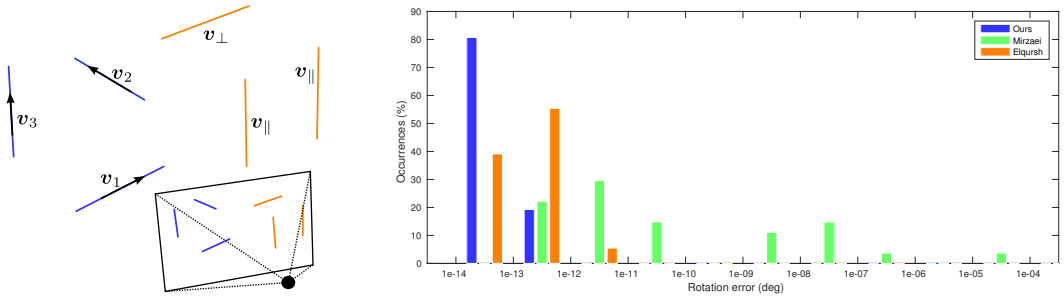
5.1 Minimal solutions

A minimal solution provides the exact solution (or multiple mathematically feasible solutions) from a minimal set of input data. However, even though a minimal solution took ideal data unaffected by noise, working under finite precision arithmetic could still provoke numerical issues that can deteriorate the performance introducing some degree of error.

For this reason, we use here ideal data to test our proposed minimal solution and other related minimal alternatives using ideal, error-free data as input in order to extract meaningful conclusions on the numerical behaviour and stability of the solutions.

As already shown in some previous work [13], there are two distinct minimal configurations of lines in a Manhattan world from which the Manhattan directions can be recovered:

1. The three lines are orthogonal to each other (see Figure 1(a) or Figure 7(a) in blue): This is the case of the P3oA problem addressed in this work, also considered by Mirzaei and Roumeliotis in [13].



(a) Minimal cases: 3 orthogonal lines (blue), 2 par. lines + 1 ort. line (orange)

(b) Histogram of numerical errors

Fig. 7 Numerical evaluation of minimal solutions. (a) Synthetic data: A P3oA minimal configuration (3 orthogonal lines, in blue) and a triplet formed by 2 parallel lines and 1 orthogonal line (orange) (b)(c) The histogram (for 10^6 samples) of the numerical errors shows higher numerical stability for our approach compared to the solution of P3oa with the alternative solver [13] (green) or the solution of the *2 par. + 1 ort.* minimal problem with [15] (orange)

2. Two lines are parallel, and the third line is orthogonal to them (see Figure 7(a) in orange): This is the case exploited by Elqursh and Elgammal in [15].

To obtain the necessary evaluation data we assume a ideal camera with a horizontal Field of View (FOV) of 50° and a 4:3 aspect ratio. We generated then 10^6 random triplets of lines (asserting they lie inside the camera's field of view) for both of the minimal problems mentioned above. Since multiple solutions arise for the minimal cases only the one corresponding to the ground truth is kept for evaluation.

In this section we are going to show:

- The differences between *minimal solvers* for the P3oA problem. This minimal problem can be solved more accurately and faster using our proposed solution than with the alternative solution presented in [13].
- The differences between the two *minimal problems* connected to the Manhattan world, using our solver for the P3oA problem and the simple minimal solution in [15] for the *2 parallels + 1 orthogonal* configuration.

5.1.1 Comparison of the P3oA solvers

For the same set of 10^6 random P3oA problems, we compare the performance of our proposed minimal solution in Algorithm 2 to the minimal solution provided by Mirzaei and Roumeliotis [13].

The evaluation results depicted in Figure 7(b) show that our minimal solution is more accurate and numerically stable than the Mirzaei-Roumeliotis' alternative, with a difference of several *orders of magnitude* in the numerical error. Furthermore, our approach is also more computationally efficient: The average run time for our Matlab implementation is 0.9 ms whereas the mean time for Mirzaei-Roumeliotis' is of 8.2 ms, that is, almost an order of magnitude lower.

This could be expected for two main reasons. On the one hand, our solution directly addresses the particular case of the P3oA problem, exploiting the intrinsic characteristics of the configuration. On the other hand, the minimal solution in [13] stems from the particularization of a closed-form solution to the estimation of Manhattan world directions from an arbitrary number of observed lines. This is a more complex problem, and so is also its solver. As a result, the particularization for the minimal problem, although simpler, still keeps part of such complexity. This behaviour can be also explained from a more practical point of view looking at the operations involved: our minimal solution relies solely on dot and vector products, and at some point the eigenvalue decomposition of a 2×2 matrix. Mirzaei-Roumeliotis's method, however, needs to perform the eigenvalue decomposition of a 8×8 matrix, among other operations. Since the decomposition of the matrix becomes worse conditioned for bigger matrices, this may also explain the observed numerical behaviour.

5.1.2 Comparison of the minimal problems

We assess the numerical behaviour of the solutions to the minimal problem of orientation from lines in a Manhattan world in its two possible forms. For that, we solved 10^6 random minimal cases of orthogonal lines with our approach (in Algorithm 2) and other 10^6 random cases of 2 parallel + 1 orthogonal lines with the minimal solution proposed by Elqursh and Elgammal [15].

We observe from the results in Figure 7(b) that, although Elqursh-Elgammal's solution performs well, ours does better with a typical numerical error one order of magnitude lower.

The numerical complexity of Elqursh-Elgammal's minimal solution is similar, or even lower, than our algorithm: Just linear operations, mainly dot and vector products. Hence, we guess that the lower numerical stability of Elqursh-Elgammal's solution stems from a higher occurrence rate of near-degenerate cases in the *2 parallels + orthogonal* minimal problem than in the *3 orthogonals* case. This suggests, at least qualitatively, that the exploitation of the P3oA problem may provide higher robustness.

5.2 Visual gyroscope

The proposed approach for the estimation of relative rotation between frames (Algorithm 3) can be used as a *visual gyroscope*. A commonly used approach for a line-based visual gyroscope resorts on the extraction of the Vanishing Points (VP) in each image and the computation of a rotation from those VPs. An alternative to these *Vanishing Point*-based (VP-based) methods are the *Minimal Solution*-based (MS-based) methods, such as ours or Elqursh-Elgammal's in [15]. These do not rely on the computation of the scene VPs but on exploiting the existence of any minimal Manhattan configuration. The pipelines of our proposal and that of Elqursh-Elgammal are very similar, being the main difference the particular minimal solution upon which the method is built. This is more in-depth analyzed next.

5.2.1 Comparison of the Minimal Solution*-based methods*

Here, we show a series of experiments aimed at assessing the performance of our method against the one reported by Elqursh and Elgammal in [15] under different conditions and variables.

To make the analysis and conclusions clearer, we choose a very simple testing scenario, consisting of a scene containing just one cube, which provides a controlled amount of lines in Manhattan directions only, so configurations for both algorithms are available.

In order to have perfect, reliable groundtruth of the rotation we generate the described *cube scenario* synthetically. The metric chosen for error computation is again the rotation angle defined in (34).

To generate the synthetic data we simulate a 1m × 1m × 1m cube (see Figure 8(a)) and the camera is randomly placed at distances below 9 m. We assert that the entire cube is projected onto the image and that, for each pair of images, the same lines are simultaneously observed.

The simulated camera (with no distortion) is characterized by the calibration matrix

$$\boldsymbol{K} = \begin{bmatrix} 700 & 0 & 320 \\ 0 & 700 & 240 \\ 0 & 0 & 1 \end{bmatrix} \tag{41}$$

which provides a horizontal field of view (FOV) of 49°.

The results, derived from a population of 10^4 samples, are depicted graphically by means of boxplot diagrams: the central line is the median and the lower and upper borders of the box represent the 25th and 75th percentiles, respectively.

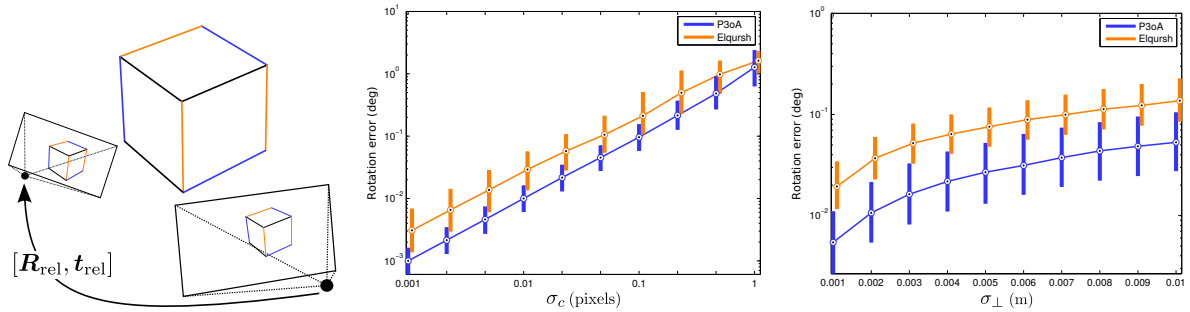
Image noise resilience This error is due to camera noise and image-related artifacts that affect the precision of the input line data. We simplify all the factors affecting the data quality into a single random noise added to the image coordinates of the segment end-points. The noise is modeled by a Gaussian distribution with zero mean and an increasing standard deviation σ_c.

The results depicted in Figure 8(b) exhibit a linear trend between the order of noise in the image and the order of the rotation error. For the whole range of noise levels considered, our method proves to be more precise.

Manhattan assumption The set of lines in real structured scenes are not exactly parallel or orthogonal, so the Manhattan assumption is violated. Therefore, we also test the influence of deviations from the Manhattan assumption in the scene model.

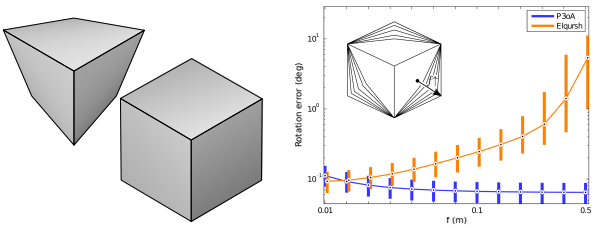
In this experiment images are noise-free, but an increasing level of Gaussian noise $\sigma_\perp$ is applied to the 3D points of the cube (before projection). The immediate effect is that the orthogonality or parallelism among 3D segments is no longer fulfilled. Values for $\sigma_\perp$ spanning from 1 mm up to 1 cm were used and it can be seen in Figure 8(c) that both methods still yield good results. So it can be concluded that both methods stand bounded deviations from the Manhattan assumption although, once again, our approach produces slightly better results for all the considered noise range.

Perspective distortion effect One important characteristic of the minimal solution exploited by Elqursh and Elgammal is that it internally depends on the computation of a Vanishing Point from two lines. The precision achievable when computing vanishing points strongly depends on the distortion arising from the scene perspective projection, which maps points lying at infinity (namely the vanishing points) to finite points. The weaker the perspective effect gets the farther the vanishing points lie, becoming more sensitive to image noise [27]. This simple effect is depicted in Figure 9(a) for our



(a) Setup (b) Image noise (c) Structure noise

Fig. 8 Synthetic experiments on the noise resilience of our approach and Elqursh-Elgammal's [15]. (a) The setup for the test. The cube is observed from two poses and the relative transformation is solved. (b)(c) Our method outperforms Elqursh-Elgammal's for all noise levels (10^4 samples)



(a) (b)

Fig. 9 The perspective distortion effect greatly depends on the FOV of the camera. (a) Perspective distortion in large FOV (left) and reduced FOV (right) camera. (b) Our approach is noticeably more robust that Elqursh-Elgammal's [15] under weak perspective

testing environment. On the other hand, the solution to the P3oA problem used in our approach does not rely on vanishing points at all.

As a result the performance of both approaches should vary with the degree of perspective distortion introduced by the projective camera. In order to assess this effect we deploy an experiment in which focal distance and camera zoom vary accordingly to modify perspective deformation of the observed cube while keeping the object size in the image constant. Figure 9(b) shows how as the perspective deformation becomes weaker for increasing focal length the results provided by Elqursh-Elgammal's algorithm get worse while ours even improve. However, for very small focal distances corresponding to wide FOV cameras Elqursh's approach can be slightly more precise than ours. This hints that it could be fruitful to address a simultaneous exploitation of both approaches in order to get a more robust method for any kind of situation.

Verification of conclusions with real data Finally, to contrast the results and conclusions obtained from the synthetic experiments, we tested the same methods in the *cabinet* dataset of the TUM Benchmark [33]. The scene used for this dataset is very similar to that of our previous synthetic framework. Namely, the dataset consists of a continous video sequence of a plain cabinet which is recorded with a Kinect device from different viewpoints. The absolute pose groundtruth of the Kinect device for each frame is provided with the dataset, so that we can still evaluate the rotation error quantitatively.

The borders of the cabinet have been detected and matched for all the images of the sequence. Some of the pairs used in this experiment are shown in Figures 10(a) and 10(b). We use superposition with false color to visualize the pair of images simultaneously.

The statistics of the committed error, plotted in Figures 10(c) and 10(d) and detailed in Table 1, confirm the conclusion also reached from the synthetic results: Our algorithm tends to perform better than Elqursh-Elgammal's, with higher precision and robustness.

Table 1 Statistics of error in the *cabinet* dataset [33]

Algorithm	$\boldsymbol{R}$ estimation error (deg)			
	Mean	q_1	Median	q_3
Ours	0.655	0.466	0.623	0.846
Elqursh [15]	0.789	0.503	0.715	0.960

5.2.2 Real data evaluation

Finally, we analyze the performance of the *Minimal Solution*-based (MS-based) algorithms and a more traditional *Vanishing Point*-based (VP-based) algorithm built upon the solution provided by Mirzaei and Roumeliotis in [13].

The methods are evaluated with the ICL-NUIM datasets [34], which provide video sequences recorded in a typical structured office environment. Lines were automatically detected using the LSD detector [35] and the matching between frames was performed using the LBD descriptor [36]. Then, the relative rotation was computed

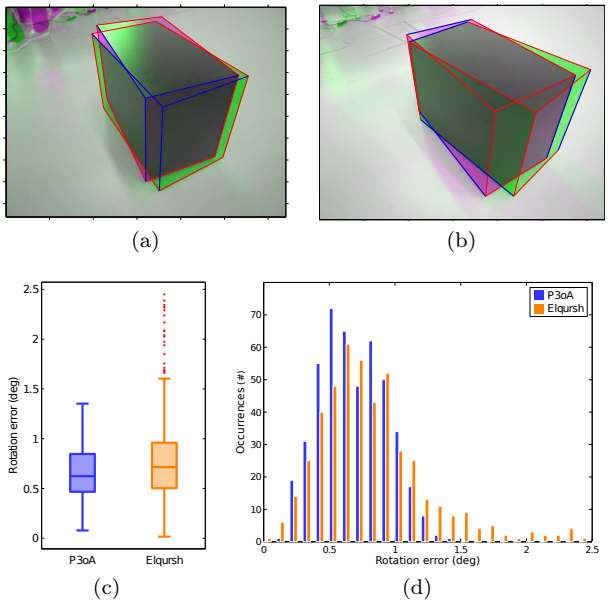


Fig. 10 Evaluation on real data. (a)(b) Two of the evaluated pairs of frames (one example of available intersecting and non intersecting orthogonal triplets marked in blue) (c)(d) Boxplot and histogram of the rotation error

between consecutive frames of the sequences and compared to the accompanying groundtruth. We calculate two different error metrics from each sequence:

- The root-mean-square error (RMSE), which due to its quadratic nature gives a good measure of the robustness. That is, if any of the committed errors becomes notably high, the RMSE grows accordingly.
- The median of the error, which is a good statistic for the general precision of the method. Since the median is much less affected by outliers, it captures the main trend of the error better than the RMSE.

The results in Tables 2(a) and 2(b) show that our proposal outperforms the compared methods, achieving higher precision and robustness than Elqursh-Elgammal's [15], which agrees with the conclusions reached in all the previous experiments.

As for the VP-based methods, they are expected to perform well if the lines supporting the three orthogonal vanishing points linked to a Manhattan world framework can be clearly clustered (see Figure 11(a)). This explains why, in some cases, the median of the error is lower for Mirzaei-Roumeliotis's method (see Table 2(b)). However, when there is no set of Manhattan directions with a clear set of lines supporting it, the performance of this approach deteriorates and the classification of the lines into Manhattan directions tends to fail (see Figures 11(b) and 11(c)). This case is likely to happen when the number of lines lowers, when there are many more lines outside the Manhattan framework than inside, or the environment, although structured, is not well modeled by the Manhattan assumption (see Atlanta world [37] or the mixture of Manhattan frames [38]). This trait of the VP-based methods makes them more prone to fail catastrophically, driving them to higher estimation errors. This justifies the significantly high RMSE in Table 2(a) obtained for Mirzaei-Roumeliotis' approach in the tested sequences.

Table 2 Metrics of the estimation error in ICL-NUIM datasets [34]

(a) RMSE of the rotation error

	of kt0	of kt1	of kt2	of kt3
Ours	**0.557**	**0.447**	**1.151**	**0.416**
Elqursh-Elgammal [15]	0.585	0.719	1.505	0.420
Mirzaei-Roumeliotis [13]	2.740	5.959	4.598	3.549

(b) Median of the rotation error

	of kt0	of kt1	of kt2	of kt3
Ours	**0.317**	**0.231**	0.459	**0.232**
Elqursh-Elgammal [15]	0.365	0.292	1.071	**0.232**
Mirzaei-Roumeliotis [13]	0.320	0.323	**0.416**	0.357

6 Conclusions

In this paper it has been presented a new closed-form solution for the Perspective-3-Angles (P3A) problem in the particular case that all angles are orthogonal (P3oA problem). This solution can solve any configuration of 3D orthogonal directions, both intersecting and non-intersecting lines. A purelly algebraic approach is derived to solve the most simple situation of meeting lines and then the solution is expanded to solve the most general case of non-meeting lines. The resulting solution is both efficient and numerically stable, as validated experimentally.

Secondly, a minimal solution has been presented for the estimation of the relative rotation between two cameras based on the solution to the P3oA problem. This minimal solution has proven exploitable in a RANSAC framework to classify triplets of lines fulfilling the orthogonality constraint and a robust global solution using all available triplets has been also proposed. The performance of this approach has been evaluated with both synthetic and real data.

(a) Rich: VP Success (b) Poor: VP Success (c) Poor: VP Fail

Fig. 11 Examples of rich and poor Manhattan scenarios in the sequences of [34]. (a) The VP-based methods perform well in clearly structured environments where the Manhattan directions are well supported. (b)(c) However, they easily fail when there is no clear support for the Manhattan directions.

A Reduction of pencil basis parameter

The bilinear map $\boldsymbol{B}_{ij}$ is greatly simplified when pencil bases are used. The substitution of $\boldsymbol{\Omega}_k$ defined in (7) into (5) gives

$$\begin{aligned}\boldsymbol{B}_{ij} &= \underbrace{\begin{bmatrix}\boldsymbol{t} & \boldsymbol{t}\times\boldsymbol{n}_i\end{bmatrix}^\top}_{2\times 3}\underbrace{\begin{bmatrix}\boldsymbol{t} & \boldsymbol{t}\times\boldsymbol{n}_j\end{bmatrix}}_{3\times 2}\\ &= \begin{bmatrix}\boldsymbol{t}^\top\boldsymbol{t} & \boldsymbol{t}^\top(\boldsymbol{t}\times\boldsymbol{n}_j)\\ (\boldsymbol{t}\times\boldsymbol{n}_i)^\top\boldsymbol{t} & (\boldsymbol{t}\times\boldsymbol{n}_i)^\top(\boldsymbol{t}\times\boldsymbol{n}_j)\end{bmatrix}\\ &= \begin{bmatrix}1 & 0\\ 0 & -\alpha_{ij}\end{bmatrix}\end{aligned}$$

since $\|\boldsymbol{t}\| = 1 \Rightarrow \boldsymbol{t}^\top\boldsymbol{t} = 1$ and by definition of the cross product $\boldsymbol{t}^\top(\boldsymbol{t}\times\boldsymbol{n}) = 0$ for any vector $\boldsymbol{n}$.

The remaining parameter

$$\alpha_{ij} = -(\boldsymbol{t}\times\boldsymbol{n}_i)^\top(\boldsymbol{t}\times\boldsymbol{n}_j)$$

can be further simplified to a single scalar product by applying some properties of the cross product. Firstly, the skew matrix representation for the cross product as well as the property $[\boldsymbol{t}]_\times^\top = -[\boldsymbol{t}]_\times$ allows us to write

$$\alpha_{ij} = \boldsymbol{n}_i^\top[\boldsymbol{t}]_\times[\boldsymbol{t}]_\times\boldsymbol{n}_j$$

and then rewriting the product of skew matrices in the equivalent form

$$[\boldsymbol{a}]_\times[\boldsymbol{b}]_\times = \boldsymbol{b}\boldsymbol{a}^\top - (\boldsymbol{a}\cdot\boldsymbol{b})\mathbf{I}_3$$

the expression finally reduces to

$$\alpha_{ij} = \boldsymbol{n}_i^\top\left(\boldsymbol{t}\boldsymbol{t}^\top - (\boldsymbol{t}^\top\boldsymbol{t})\cdot\mathbf{I}_3\right)\boldsymbol{n}_j = -\boldsymbol{n}_i^\top\boldsymbol{n}_j$$

where it is used that $\boldsymbol{t}\perp\boldsymbol{n}_k$ by the definition in (6).

B Relations between interpretation planes for the degenerate P3oA problem

Given a trihedron formed by three intersecting orthogonal lines, if the projection of two lines $\mathbf{l}_i$ and $\mathbf{l}_j$ become the same, the normals of the interpretation planes of the lines fulfill $\boldsymbol{n}_i \parallel \boldsymbol{n}_j \perp \boldsymbol{n}_k$.

The corresponding proof is developed in a sequence of minor steps:

- If the lines $\mathbf{l}_i$ and $\mathbf{l}_j$ are parallel, so are the corresponding normals $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$: $\mathbf{l}_i \parallel \mathbf{l}_j \Rightarrow \boldsymbol{n}_i \parallel \boldsymbol{n}_j$.
- If the normals $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$ are parallel, the camera center must lie in the IJ plane defined by the lines L_i and L_j.
- If the camera center lies in the IJ plane, the normal $\boldsymbol{n}_k$ is orthogonal to $\boldsymbol{n}_i = \boldsymbol{n}_j$.

B.1 Parallelism of $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$

Let $\mathbf{l}_i$ and $\mathbf{l}_j$ be the homogeneous vectors corresponding to the projection of the lines into the camera image. Since the projective entities $\mathbf{l}_i, \mathbf{l}_j \in \mathbb{P}^2$ are defined up to scale, we use the similarity operator $\mathbf{l}_i \sim \mathbf{l}_j$ to represent the equality of both variables in the projective space $\mathbb{P}^2$. Two vectors are then equivalent if they are parallel, or stated otherwise,

$$\mathbf{l}_i \times \mathbf{l}_j = \mathbf{0}.$$

The homogeneous lines in the image are related to the normal of the corresponding interpretation plane through the intrinsic calibration matrix $\boldsymbol{K}$ [27]

$$\boldsymbol{n} \sim \boldsymbol{K}^\top\mathbf{l},$$

so the equivalency relation is transmitted to the normals:

$$\begin{aligned}\boldsymbol{n}_i \times \boldsymbol{n}_j &= (\boldsymbol{K}^\top\mathbf{l}_i)\times(\boldsymbol{K}^\top\mathbf{l}_j)\\ &= \det(\boldsymbol{K})\boldsymbol{K}^{-\top}(\mathbf{l}_i\times\mathbf{l}_j) = \mathbf{0} \Rightarrow \boldsymbol{n}_i \sim \boldsymbol{n}_j\end{aligned}$$

B.2 Position of the camera to fulfill $\boldsymbol{n}_i \parallel \boldsymbol{n}_j$

Let us assume, without loss of generality, that the three intersecting lines fit the axes of the canonical coordinate system. As a result, the direction of the lines in this particular case are given by the canonical vectors $\{\boldsymbol{e}_k\}_{k=1}^3$. Denote the position and orientation of the camera as seen from this coordinate system as $\boldsymbol{t}$ and $\boldsymbol{R}$, respectively. The normal to the intepretation plane of each line L_k, as seen from the camera, is then equal (up to scale) to

$$\boldsymbol{n}_k \sim \boldsymbol{R}^\top(\boldsymbol{e}_k\times\boldsymbol{t}). \tag{42}$$

From the equivalency of the lines i and j it follows then that

$$\begin{aligned}\boldsymbol{n}_i \sim \boldsymbol{n}_j &\Rightarrow (\boldsymbol{R}^\top(\boldsymbol{e}_i\times\boldsymbol{t}))\times(\boldsymbol{R}^\top(\boldsymbol{e}_j\times\boldsymbol{t}))\\ &= \boldsymbol{R}^\top((\boldsymbol{e}_i\times\boldsymbol{t})\times(\boldsymbol{e}_j\times\boldsymbol{t}))\\ &= \mathbf{0} \Rightarrow (\boldsymbol{e}_i\times\boldsymbol{t})\times(\boldsymbol{e}_j\times\boldsymbol{t}) = \mathbf{0}\end{aligned}$$

and this expression is symbolically equivalent to

$$\begin{aligned}(\boldsymbol{e}_i\times\boldsymbol{t})\times(\boldsymbol{e}_j\times\boldsymbol{t}) &= [\boldsymbol{e}_i\times\boldsymbol{t}]_\times[\boldsymbol{e}_j]_\times\boldsymbol{t}\\ &= (\boldsymbol{t}\boldsymbol{e}_i^\top - \boldsymbol{e}_i\boldsymbol{t}^\top)[\boldsymbol{e}_j]_\times\boldsymbol{t}\\ &= (\boldsymbol{e}_k^\top\boldsymbol{t})\,\boldsymbol{t}\end{aligned}$$

So, it is concluded that the equivalency $\boldsymbol{n}_i \sim \boldsymbol{n}_j$ is only possible if $\boldsymbol{t} = \boldsymbol{0}$ or $\boldsymbol{e}_k^\top \boldsymbol{t} = 0$. The first solution forces the camera to be in the point of intersection of the three lines, but this makes no sense. The second solution implies that, for L_i and L_j to project into the same image line $\mathbf{l}_i \sim \mathbf{l}_j$, the camera center must lie in the plane defined by lines L_i and L_j.

B.3 Orthogonality of normals when the camera lies in the IJ plane

Now, we will prove that if the camera center lies in the IJ plane, the normal of the interpretation plane for the remaining line L_k is orthogonal to both $\boldsymbol{n}_i \sim \boldsymbol{n}_j$. Say $\boldsymbol{n}_k$ and $\boldsymbol{n}_i$ are orthogonal, which is equivalent to state $\boldsymbol{n}_k^\top \boldsymbol{n}_i = 0$. Using the relation in (42)

$$\begin{aligned}\boldsymbol{n}_k^\top \boldsymbol{n}_i &= (\boldsymbol{e}_k \times \boldsymbol{t})\boldsymbol{R}\boldsymbol{R}^\top(\boldsymbol{e}_i \times \boldsymbol{t}) \\ &= \boldsymbol{t}^\top [\boldsymbol{e}_k]_\times^\top [\boldsymbol{e}_i]_\times \boldsymbol{t} \\ &= -\boldsymbol{t}^\top(\boldsymbol{e}_i\boldsymbol{e}_k^\top - (\boldsymbol{e}_k^\top \boldsymbol{e}_i)\mathbf{I}_3)\boldsymbol{t} \\ &= -(\boldsymbol{t}^\top \boldsymbol{e}_i)(\boldsymbol{e}_k^\top \boldsymbol{t}) \\ &= -(\boldsymbol{e}_i^\top \boldsymbol{t})(\boldsymbol{e}_k^\top \boldsymbol{t}) = 0\end{aligned}$$

and the condition above is then fulfilled only if the camera center lies in the JK plane, the IJ plane, or both. Similarly, if $\boldsymbol{n}_k$ and $\boldsymbol{n}_j$ are orthogonal (necessary from $\boldsymbol{n}_i \sim \boldsymbol{n}_j$) the camera center lies in the IJ plane, the IK plane, or both. As a result, we see that if the camera lies in the IJ plane as assumed, the orthogonality constraints are fulfilled.

In conclusion we see that if the projection of two lines, $\mathbf{l}_i$ and $\mathbf{l}_j$, are coincident then the camera center lies in the plane formed by L_i and L_j. This, at the same time, provokes that $\boldsymbol{n}_k \perp \boldsymbol{n}_i$, or equivalently, $\boldsymbol{n}_k \perp \boldsymbol{n}_j$.

C Reparameterization of radical equation as a quadratic form

The equation

$$\boldsymbol{n}_k^\top \boldsymbol{t} \cdot \bar{\alpha} + \alpha_{ij} \cdot \boldsymbol{n}_k^\top [\boldsymbol{t}]_\times \boldsymbol{n}_* = 0$$

defined in (18) is non-linear wrt $\boldsymbol{n}_*$ due to the square root operation in

$$\begin{aligned}\bar{\alpha} &= \pm\sqrt{\alpha_{*i}\alpha_{ij}\alpha_{j*}} \\ &= \pm\sqrt{-\left(\boldsymbol{n}_*^\top \boldsymbol{n}_i\right)\left(\boldsymbol{n}_i^\top \boldsymbol{n}_j\right)\left(\boldsymbol{n}_j^\top \boldsymbol{n}_*\right)}\end{aligned}$$

which makes the equation radical. As usual for these equations, we separate both terms in the sum and square them to get an almost-equivalent quadratic equation

$$\begin{aligned}\left(\boldsymbol{n}_k^\top \boldsymbol{t}\right)^2 \cdot (\pm\sqrt{\alpha_{*i}\alpha_{ij}\alpha_{j*}})^2 &= \alpha_{ij}^2 \cdot (\boldsymbol{n}_k^\top [\boldsymbol{t}]_\times \boldsymbol{n}_*)^2 \\ \frac{\left(\boldsymbol{n}_k^\top \boldsymbol{t}\right)^2}{\alpha_{ij}} \cdot \alpha_{*i}\alpha_{j*} &= (\boldsymbol{n}_k^\top [\boldsymbol{t}]_\times \boldsymbol{n}_*)^2\end{aligned}$$

Then, we substitute α_{*i} and α_{j*} and arrange the matrix operations

$$\begin{aligned}\frac{(\boldsymbol{n}_k^\top \boldsymbol{t})^2}{\alpha_{ij}} \cdot \boldsymbol{n}_*^\top \left(\boldsymbol{n}_i\boldsymbol{n}_j^\top\right)\boldsymbol{n}_* &= (\boldsymbol{n}_k^\top [\boldsymbol{t}]_\times \boldsymbol{n}_*)^\top \boldsymbol{n}_k^\top [\boldsymbol{t}]_\times \boldsymbol{n}_* \\ &= \boldsymbol{n}_*^\top [\boldsymbol{t}]_\times^\top \left(\boldsymbol{n}_k\boldsymbol{n}_k^\top\right)[\boldsymbol{t}]_\times \boldsymbol{n}_*\end{aligned}$$

so that the condition above can be encoded by a quadratic form

$$\boldsymbol{n}_*^\top \left(\frac{(\boldsymbol{n}_k^\top \boldsymbol{t})^2}{\alpha_{ij}} \cdot \boldsymbol{n}_i\boldsymbol{n}_j^\top - [\boldsymbol{t}]_\times^\top \boldsymbol{n}_k\boldsymbol{n}_k^\top [\boldsymbol{t}]_\times\right)\boldsymbol{n}_* = 0$$

From all infinite matrix representations available for the quadratic form given above we choose the symmetric one

$$\boldsymbol{Q} = \frac{1}{2}\frac{(\boldsymbol{n}_k^\top \boldsymbol{t})^2}{\alpha_{ij}}\left(\boldsymbol{n}_i\boldsymbol{n}_j^\top + \boldsymbol{n}_j\boldsymbol{n}_i^\top\right) + [\boldsymbol{t}]_\times \left(\boldsymbol{n}_k\boldsymbol{n}_k^\top\right)[\boldsymbol{t}]_\times$$

which permits us to readily diagonalize the quadratic form by eigenvalue decomposition in a unique way.

Then, the original problem in (18) is equivalent to solving

$$\boldsymbol{n}_*^\top \boldsymbol{Q}\, \boldsymbol{n}_* = 0$$

for the defined $\boldsymbol{Q}$.

D Necker duality and parallax effect

The two distinct solutions of the P3oA problem for the case of meeting lines are mirrored wrt the plane whose normal is the back-projection direction $\boldsymbol{t}$ through the intersection point. This relation can be easily expressed through a reflection or Householder matrix [27]. Let $\boldsymbol{V}$ be the real solution and $\boldsymbol{V}^*$ the mirrored one. The following relation stands:

$$\boldsymbol{V} = \boldsymbol{H}_{\boldsymbol{t}}\, \boldsymbol{V}^* \tag{43}$$

with $\boldsymbol{H}_{\boldsymbol{t}} = \mathbf{I}_3 - 2\,\boldsymbol{t}\,\boldsymbol{t}^\top/\boldsymbol{t}^\top\boldsymbol{t}$. This relation only stands when the three lines meet in a single point, otherwise there exists no reflection fulfilling the relation (43), even though this duality still exists (see green configuration in Figure 4).

Without loss of generality, we will use the case of meeting lines to prove that both dual solutions provide the same rotation in the case of zero baseline (pure rotation). In such cases, taking the pair of false configurations will produce the relative rotation

$$\begin{aligned}\boldsymbol{R}^* &= \boldsymbol{V}_1^* (\boldsymbol{V}_2^*)^\top \\ &= \boldsymbol{H}_{p_1}\boldsymbol{V}_1\, \boldsymbol{V}_2^\top \boldsymbol{H}_{p_2}^\top \\ &= \boldsymbol{H}_{p_1}\, \boldsymbol{R}\, \boldsymbol{H}_{p_2}^\top\end{aligned} \tag{44}$$

where p_i stands for the 3D coordinates of intersection point in i-th image wrt the camera frame. A rigid transformation exists between p_1 and p_2

$$p_1 = \boldsymbol{R}\, p_2 + \boldsymbol{t}_{\text{rel}}$$

so that, in the case of zero baseline we have

$$\boldsymbol{H}_{p_1} = \mathbf{I} - 2\frac{\boldsymbol{R}p_2 p_2^\top \boldsymbol{R}^\top}{p_2^\top p_2} = \boldsymbol{R}\boldsymbol{H}_{p_2}\boldsymbol{R}^\top$$

and substituting in expression (44) reveals that

$$\boldsymbol{R}^* = \boldsymbol{R}\boldsymbol{H}_{p_2}\boldsymbol{R}^\top\, \boldsymbol{R}\, \boldsymbol{H}_{p_2}^\top = \boldsymbol{R}$$

As a conclusion, in the case of pure rotation the true and false solutions are the same. So, the greater the parallax effect due to non zero baseline $\|\boldsymbol{t}_{\text{rel}}\|$, the bigger the difference between both possible solutions.

Acknowledgements We would like to thank to Ali Elqursh for providing us the source code of the odometry method of Elqursh and Elgammal [15].

This work has been supported by two projects: "GiraffPlus", funded by EU under contract FP7-ICT-#288173, and "TAROTH: New developments toward a robot at home", funded by the Spanish Government and the "European Regional Development Fund ERDF" under contract DPI2011-25483.

References

1. B. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International Journal of Computer Vision*, vol. 13, no. 3, pp. 331–356, 1994.
2. V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, pp. 155–166, July 2008.
3. J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (DLS) method for PnP," in *IEEE International Conference on Computer Vision (ICCV), 2011*, pp. 383–390, IEEE, 2011.
4. S. Li, C. Xu, and M. Xie, "A robust O(n) solution to the perspective-n-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1444–1450, 2012.
5. F. M. Mirzaei and S. I. Roumeliotis, "Globally optimal pose estimation from line correspondences," in *IEEE International Conference on Robotics and Automation (ICRA), 2011*, pp. 5581–5588, Ieee, May 2011.
6. L. Zhang, C. Xu, K.-M. Lee, and R. Koch, "Robust and efficient pose estimation from line correspondences," in *Asian Conference on Computer Vision (ACCV), 2012*, pp. 217–230, Springer, 2013.
7. S. Ramalingam, S. Bouaziz, and P. Sturm, "Pose estimation using both points and lines for geo-localization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 4716–4723, IEEE, 2011.
8. R. I. Hartley, "Lines and points in three views and the trifocal tensor," *International Journal of Computer Vision*, vol. 22, no. 2, pp. 125–140, 1997.
9. A. Bartoli and P. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment," *Computer Vision and Image Understanding*, vol. 100, no. 3, pp. 416–441, 2005.
10. J. Košecká and W. Zhang, "Extraction, matching, and pose recovery based on dominant rectangular structures," *Computer Vision and Image Understanding*, vol. 100, no. 3, pp. 274–293, 2005.
11. G. Schindler, P. Krishnamurthy, and F. Dellaert, "Line-based structure from motion for urban environments," in *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 846–853, IEEE, 2006.
12. W. Förstner, "Optimal vanishing point detection and rotation estimation of single images from a legoland scene," in *ISPRS Commission III Symposium of Photogrammetric Computer Vision and Image Analysis*, pp. 157–162, 2010.
13. F. M. Mirzaei and S. I. Roumeliotis, "Optimal estimation of vanishing points in a Manhattan world," in *IEEE International Conference on Computer Vision (ICCV), 2011*, pp. 2454–2461, IEEE, 2011.
14. J. C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys, "Globally optimal line clustering and vanishing point estimation in Manhattan world," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 638–645, IEEE, 2012.
15. A. Elqursh and A. Elgammal, "Line-based relative pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011*, pp. 3049–3056, IEEE, 2011.
16. M. Finotto and E. Menegatti, "Humanoid gait stabilization based on omnidirectional visual gyroscope," in *Workshop on Humanoid Soccer Robots (Humanoids' 09)*, 2009.
17. E. Rondon, L. R. G. Carrillo, and I. Fantoni, "Vision-based altitude, position and speed regulation of a quadrotor rotorcraft," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010*, pp. 628–633, 2010.
18. S. T. Barnard, "Choosing a basis for perceptual space," *Computer vision, graphics, and image processing*, vol. 29, no. 1, pp. 87–99, 1985.
19. K. I. Kanatani, "Constraints on length and angle," *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 1, pp. 28–42, 1988.
20. Y. Wu, S. S. Iyengar, R. Jain, and S. Bose, "A new generalized computational framework for finding object orientation using perspective trihedral angle constraint," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, pp. 961–975, 1994.
21. M. Dhome, M. Richetin, J.-T. Lapreste, and G. Rives, "Determination of the attitude of 3d objects from a single perspective view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1265–1278, 1989.
22. A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, 2000.
23. M. E. Antone and S. Teller, "Automatic recovery of relative camera rotations for urban scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2000.*, vol. 2, pp. 282–289, IEEE, 2000.
24. J. Košecká and W. Zhang, "Video compass," in *European Conference on Computer Vision (ECCV), 2002*, pp. 476–490, Springer, 2002.
25. P. Denis, J. Elder, and F. Estrada, *Efficient edge-based methods for estimating manhattan frames in urban imagery.* Springer, 2008.
26. J. C. Bazin and M. Pollefeys, "3-line RANSAC for orthogonal vanishing point detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012*, pp. 4282–4287, IEEE, 2012.
27. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* New York, NY, USA: Cambridge University Press, 2 ed., 2003.
28. D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.
29. R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *International Journal of Computer Vision*, vol. 103, no. 3, pp. 267–305, 2013.
30. L. Carlone and F. Dellaert, "Duality-based verification techniques for 2D SLAM," in *Intl. Conf. on Robotics and Automation (ICRA)*, 2015.
31. J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*, vol. 3. Oxford University Press Oxford, 2004.
32. K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 5, pp. 698–700, 1987.
33. J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012*, pp. 573–580, IEEE, 2012.
34. A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, (Hong Kong, China), May 2014.
35. R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a Line Segment Detector," *Image Processing On Line*, vol. 2, pp. 35–55, Mar. 2012.
36. L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.
37. G. Schindler and F. Dellaert, "Atlanta world: an expectation maximization framework for simultaneous low-level edge

grouping and camera calibration in complex man-made environments," *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 203–209, 2004.

38. J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher, "A mixture of manhattan frames: Beyond the manhattan world," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 3770–3777, IEEE, 2014.