Hybrid and Non-minimal Planar Motion Estimation from Point Correspondences

Juan C. Dibene and Enrique Dunn

Stevens Institute of Technology
{jdibenes, edunn}@stevens.edu

Abstract. We address the problem of relative camera pose estimation in the context of planar motion, where the rotation axis and translation vectors are orthogonal to each other. For such scenarios, it is common to assume a known motion plane to leverage the reduced algebraic structure and geometric parameterization of the ensuing epipolar constraints. In this work, we focus on the general prior-free case, in which no assumptions about the plane of motion are made. While current solvers estimate planar motion from homogeneous (i.e. 2D-2D or 3D-3D) point correspondences, leveraging hybrid (i.e. combinations of 2D-2D, 2D-3D, and 3D-3D) point correspondences remains an open problem. We explore the solution space for the general planar motion problem and propose three novel minimal solvers from hybrid point correspondences, as well as a triplet of new non-minimal solvers from 2D-2D point correspondences bridging the theoretical gap from minimal to linear solutions. Experiments on both synthetic data and standard benchmark sequences of real-world imagery demonstrate that our proposed solvers can provide better pose estimates than homogeneous planar motion solvers (with or without motion plane prior), while achieving competitive run times.

Keywords: General Planar Motion · Camera Pose Estimation · Minimal Hybrid Solvers · Non-Minimal Solvers

1 Introduction

The problem of calibrated camera motion estimation from point observations entails the analysis and solution of the algebraic structures describing the geometry of convergent viewing rays; commonly under the pinhole camera model. Associating viewing rays among two images yields a relative pose estimation instance (i.e. unknown translation scale), while absolute pose entails either associating viewing rays to 3D points, or performing 3D alignment among corresponding 3D points defined in different frames. Commonly, solutions are derived from sets of pure 2D-2D or 2D-3D observations. However, recent works considering sets of mixed 2D-2D and 2D-3D observations have demonstrated promising results.

Hybrid solvers combining observations of heterogeneous scope (e.g. 2D-2D and 2D-3D) were explored by Josephson et al. [15] for visual localization, within the context of incremental SfM in Camposeco et al. [4], and in Bhayani et al. [2]

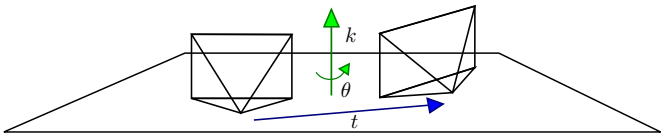

Fig. 1: Planar motion: rotation axis k orthogonal to translation t on the motion plane.

for partially calibrated semi-generalized pose. These works highlight the inherent performance trade-offs between the accurate but computationally expensive pure 2D-2D solvers (requiring more points and returning more solutions) and the more efficient but susceptible to 3D imperfections (with fewer points and solutions due to additional 3D constraints) pure 2D-3D solvers, and demonstrate that combinations of these two kinds of features may lead to better performant solvers. In principle, hybrid solvers are also applicable to egomotion estimation in the presence of heterogeneous point observations (i.e. multi-modal sensors), as it is common in autonomous mobile platforms. However, there are currently no reported minimal hybrid solvers for the general planar motion problem.

We refer by planar motion to the problem of estimating elements of $SE(3)$ where the rotation and translation vector are orthogonal to each other. Empirically, this is a common constraint applicable to cameras moving on a flat surface (e.g., mounted on indoor robots). Although the problem can be simplified by assuming a known disposition of the camera w.r.t. the motion plane, this work studies the general planar motion problem, where both the plane geometry and the in-plane motion parameters are estimated jointly. Recent work has provided minimal solvers for the 2D-2D relative pose [17] and 3D-3D absolute pose [9] cases. This work explores the space of hybrid solvers for general planar motion, identifying (and developing) a triplet of minimal problem instances. While the 2D-2D four point solver of Li et al. [17] can be considered as the planar analogue of the five point solver of Nister [20], we identified no planar analogues for the non-minimal six point solver of Philip [23] and the linear eight point solver of Longuet-Higgings [18]. Since solution uniqueness and sensitivity to depth measurement are practical concerns which can be obviated by non-minimal 2D-2D solvers, we introduce another triplet of solvers bridging the gap from an existing minimal problem instance (i.e., [17]) to a fully linear formulation.

2 Related Works

Planar Motion Estimation. The standard approach for planar motion estimation is to induce additional constraints on the motion of the camera by exploiting known priors on the camera configuration and the motion plane (e.g., optical axis parallel to motion plane) [3, 5–8, 11, 12, 21, 28] or acquiring information from additional sensors (e.g., IMU) [25–27], reducing the degrees of freedom (DOF) and number of model parameters, which in turn reduces the number of required input observations, potentially decreasing computation times while increasing estimation robustness. Choi and Park [5] introduce an algorithm for

absolute pose estimation under planar motion, assuming that the optical axis of the camera is parallel to the plane of motion. Their algorithm takes two 2D-3D point correspondences, so it also able to estimate the scale of the translation. Choi and Kim [6] propose a planar motion solver from two 2D-2D point correspondences by formulating the motion estimation problem as the finding the intersections of a line and a unit circle. Hajder and Barath [12] focus on deriving an optimal planar motion solution for the overdetermined (i.e., non-minimal) case. The two parameters of the planar motion are estimated from the roots of a sixth degree polynomial, derived from a least squares formulation using three 2D-2D point correspondences. Their solver requires that the image plane is orthogonal to the ground plane. However, the deployment of such methods is limited to particular camera/plane of motion/hardware configurations.

Planar Motion Estimation without Motion Plane Knowledge. In an orthogonal approach, Li et al. [17] present a minimal solver for planar motion from four 2D-2D point correspondences without relying on any assumptions about the plane of motion or additional sensors. Their solver leverages the structure that planar motion imposes upon the essential matrix, i.e., for planar motion, the trace of the essential matrix is zero [19]. Although more flexible than the aforementioned prior-based solvers, due to obviating extrinsic calibrations and specialized setups, the resulting solver is significantly more computationally expensive, yielding up to ten possible solutions. The work of [9] proposes a minimal solver from two 3D-3D point correspondences. Their solver proposes a geometric formulation capable of jointly estimating the plane of motion and the planar motion parameters, for a total of five DOF: two for the plane normal/axis of rotation, one for the angle of rotation, and two for the direction and scale of the translation. Although fast due to the additional constraints afforded by 3D data, their solver is limited to setups capable of 3D data acquisition (e.g., stereo cameras, depth sensors), and reports a susceptibility to depth estimation errors.

3 Hybrid and Non-minimal Solvers for Planar Motion

We propose three minimal solvers for planar motion from hybrid point correspondences, i.e., not all point correspondences are of the same type, and three non-minimal solvers from 2D-2D point correspondences: **hpc1**) hybrid solver from one 3D-3D and one 2D-3D correspondences, **hpc2**) hybrid solver from one 3D-3D and two 2D-2D correspondences, **hpc3**) hybrid solver from two 2D-3D and one 2D-2D correspondences, **nm5**) non-minimal solver from five 2D-2D correspondences, **nm6**) non-minimal solver from six 2D-2D correspondences, and **nm7**) non-minimal linear solver from seven 2D-2D correspondences. We restrict our analysis to the case of internally calibrated cameras. None of the solvers assume any knowledge of the plane of motion, relying only on the orthogonality between the axis of rotation (k) and the translation (t):

$$k^\top t = 0. \tag{1}$$

We refer to Equation 1 as the planar motion constraint. In terms of the essential matrix ($E = [t]_\times R$), the planar motion constraint can be expressed as [17, 19]:

Table 1: Top: summary of planar motion solvers. Bottom: summary of full DOF motion solvers used for experimental comparison. *: up-to-scale.

Solver	Inputs 3D-3D	2D-3D	2D-2D	Is Minimal	Is Hybrid	Plane Prior	Motion DOF	Number of Solutions
2p3d [9]	2	0	0	✓			5	1
hpc1 (ours)	1	1	0	✓	✓		5	2
hpc2 (ours)	1	0	2	✓	✓		5	4
hpc3 (ours)	0	2	1	✓	✓		5	18
4pst0 [17]	0	0	4	✓			*4	10
nm5 (ours)	0	0	5				*4	1
nm6 (ours)	0	0	6				*4	1
nm7 (ours)	0	0	7				*4	1
choi [5]	0	2	0	✓		✓	3	1
fast [6]	0	0	2	✓		✓	*2	4
lsq [12]	0	0	3			✓	*2	6
horn [14]	3	0	0	✓		N/A	6	1
lambda [22]	0	3	0	✓		N/A	6	4
nister [20]	0	0	5	✓		N/A	*5	10
longuet [18]	0	0	8			N/A	*5	1

$$\mathrm{tr}(E) = -2\sin(\alpha)(k^\top t) = 0. \tag{2}$$

where α is the rotation angle. Note, however, that $\mathrm{tr}(E) = 0$ is a necessary but not sufficient condition for planar essential matrices.

Equation 1 is the foundation of the minimal hybrid solvers, while Equation 2 is the cornerstone of the non-minimal solvers. Finally, Table 1 presents a summary of current solvers for planar motion estimation from point correspondences and our proposed solvers, elucidating the literature gap that our work fills.

3.1 HPC1: Hybrid Solver from one 3D-3D and one 2D-3D matches

Without loss of generality, one 3D-3D and one 2D-3D point correspondences give, respectively, the following constraints:

$$P_{21} = RP_{11} + t, \tag{3}$$

$$\lambda_{22}\tilde{p}_{22} = RP_{12} + t. \tag{4}$$

First, by subtracting Equation 4 from Equation 3, and taking the squared norm on both sides of the resulting equation to eliminate R (since R is an isometry), we obtain a quadratic polynomial on the unknown depth λ_{22}:

$$\|\tilde{p}_{22}\|^2\lambda_{22}^2 - 2(P_{21}\cdot\tilde{p}_{22})\lambda_{22} + \|P_{21}\|^2 - \|P_{11} - P_{12}\|^2 = 0. \tag{5}$$

Finally, we use the 3D-3D planar motion solver of [9], which leverages the orthogonality of the translation and the rotation axis, to obtain the pose (R, t) for each of the two solutions of λ_{22}.

3.2 HPC2: Hybrid Solver from one 3D-3D and two 2D-2D matches

The 3D-3D point correspondence yields the constraint expressed in Equation 3 and each 2D-2D point correspondence gives one epipolar constraint:

$$\tilde{p}_{2i}^{\top}[t]_{\times}R\tilde{p}_{1i} = 0, \quad i = \{2, 3\}. \tag{6}$$

First, by solving for t from Equation 3 and plugging the resulting expression into Equation 6, we obtain two equations in terms of R:

$$\tilde{p}_{2i}^{\top}[P_{21}]_{\times}R\tilde{p}_{1i} - \tilde{p}_{2i}^{\top}R[P_{11}]_{\times}\tilde{p}_{1i} = 0, \quad i = \{2, 3\}. \tag{7}$$

Since both equations are unaffected by the scale of R, we use Cayley's parameterization for R without the normalization term (i.e., $1/(1 + k_x^2 + k_y^2 + k_z^2)$):

$$R_h = \begin{bmatrix} 1 + k_x^2 - k_y^2 - k_z^2 & 2k_xk_y - 2k_z & 2k_xk_z + 2k_y \\ 2k_xk_y + 2k_z & 1 - k_x^2 + k_y^2 - k_z^2 & 2k_yk_z - 2k_x \\ 2k_xk_z - 2k_y & 2k_yk_z + 2k_x & 1 - k_x^2 - k_y^2 + k_z^2 \end{bmatrix}. \tag{8}$$

Then, we consider the planar motion constraint given by Equation 1. Multiplying Equation 3 by $k^{\top}$ on the left, and by the property of rotation matrices that $k^{\top}R = k^{\top}$, yields:

$$k^{\top}(P_{21} - P_{11}) = 0. \tag{9}$$

As this equation is not affected by the scale of the axis of rotation, we use:

$$k^{\top} = \begin{bmatrix} k_x & k_y & k_z \end{bmatrix}, \tag{10}$$

and Equation 9 becomes linear in the elements of k. We solve for the element with the greatest magnitude coefficient and plug it into Equation 7. Now, the problem is reduced to the intersection of two conics with up to four solutions for k. Finally, we compute $R = R_h/(1 + k_x^2 + k_y^2 + k_z^2)$ for each k and we use R to compute the corresponding t from Equation 3.

3.3 HPC3: Hybrid Solver from two 2D-3D and one 2D-2D matches

Each 2D-3D point correspondence gives an Equation 4 type constraint and the 2D-2D point correspondence gives an epipolar constraint as Equation 6. First, we solve for t for each 2D-3D constraint and substitute each into the 2D-2D constraint to obtain two equations in terms of R and the unknown depths:

$$\lambda_{2i}\tilde{p}_{23}^{\top}[\tilde{p}_{2i}]_{\times}R\tilde{p}_{13} - \tilde{p}_{23}^{\top}R[P_{1i}]_{\times}\tilde{p}_{13} = 0, \quad i = \{1, 2\}. \tag{11}$$

Using the planar motion constraint (Equation 1) and the 2D-3D constraints, we can solve for the unknown depths (recall that $k^{\top}R = k^{\top}$):

$$\lambda_{2i} = (k^{\top}P_{1i})/(k^{\top}\tilde{p}_{2i}), \quad i = \{1, 2\}, \tag{12}$$

and Equation 11 becomes:

$$(k^{\top}P_{1i})\tilde{p}_{23}^{\top}[\tilde{p}_{2i}]_{\times}R\tilde{p}_{13} - (k^{\top}\tilde{p}_{2i})\tilde{p}_{23}^{\top}R[P_{1i}]_{\times}\tilde{p}_{13} = 0, \quad i = \{1, 2\}. \tag{13}$$

Next, we obtain a third equation by subtracting the 2D-3D constraints and eliminating the unknown depths via cross product:

$$(\tilde{p}_{21} \times \tilde{p}_{22})^\top R(P_{11} - P_{12}) = 0. \tag{14}$$

All three equations are unaffected by the scale of R and k, so we again parameterize R using Equation 8 and k using Equation 10. We generate a solver for this polynomial system using GAPS [16], yielding up to 18 solutions for k. Finally, we compute $R = R_h/(1 + k_x^2 + k_y^2 + k_z^2)$ for each k and we use R to compute the corresponding t from the following linear system:

$$\begin{bmatrix} (R\tilde{p}_{13} \times \tilde{p}_{23})^\top \\ (RP_{12} \times \tilde{p}_{22})^\top \\ (\tilde{p}_{21} \times \tilde{p}_{22})^\top \end{bmatrix} t = \begin{bmatrix} 0 \\ 0 \\ -(\tilde{p}_{21} \times \tilde{p}_{22})^\top RP_{11} \end{bmatrix}. \tag{15}$$

For the sake of completeness, we now consider the case where the unknown depths of the 2D-3D constraints lie on different sides of the equations. Without loss of generality, we assume that the second 2D-3D constraint is of the form:

$$P_{22} = \lambda_{12} R\tilde{p}_{12} + t. \tag{16}$$

The derivation of Equation 13 is essentially the same. However, the third equation (Equation 14) is now:

$$(R^\top \tilde{p}_{21} \times \tilde{p}_{12})^\top P_{11} + (\tilde{p}_{21} \times R\tilde{p}_{12})^\top P_{22} = 0. \tag{17}$$

The solver generated by GAPS [16] still yields 18 solutions for k, and the computation of R and t for each k is the same as before.

3.4 NM5: Solver from five 2D-2D point matches

Each 2D-2D point correspondence gives one epipolar constraint of the form:

$$\tilde{p}_{2i}^\top E \tilde{p}_{1i} = 0, \quad i = 1, ..., 5. \tag{18}$$

Then, we identify two ways to express the trace zero constraint (Equation 2):
1) By stacking the trace zero equation into the epipolar equation matrix as:

$$\begin{bmatrix} 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1 \end{bmatrix} \tilde{E} = 0, \quad \tilde{E} = \begin{bmatrix} e_{11}\ e_{12}\ e_{13}\ e_{21}\ e_{22}\ e_{23}\ e_{31}\ e_{32}\ e_{33} \end{bmatrix}^\top. \tag{19}$$

Then, the epipolar constraints are of the form (i index suppressed for brevity):

$$\begin{bmatrix} p_2^x \tilde{p}_1^\top & p_2^y \tilde{p}_1^\top & p_2^z \tilde{p}_1^\top \end{bmatrix} \tilde{E} = 0. \tag{20}$$

Here, the superscript denotes the x, y, or z element of the corresponding vector.
2) By replacing one of the diagonal elements of E by the negated sum of the other two (e.g., $e_{11} = -(e_{22} + e_{33})$). Then, the epipolar constraints are:

$$\begin{aligned} \tilde{q}^\top \tilde{E} = 0, \quad \tilde{E} = \begin{bmatrix} e_{12}\ e_{13}\ e_{21}\ e_{22}\ e_{23}\ e_{31}\ e_{32}\ e_{33} \end{bmatrix}^\top \\ \tilde{q}^\top = \begin{bmatrix} p_1^y p_2^x, p_1^z p_2^x, p_1^x p_2^y, p_1^y p_2^y - p_1^x p_2^x, p_1^z p_2^y, p_1^x p_2^z, p_1^y p_2^z, p_1^z p_2^z - p_1^x p_2^x \end{bmatrix} \end{aligned} \tag{21}$$

As we found that formulation 2 is more resilient to noise, we use that for all three of our non-minimal solvers. Then, with five point correspondences and the trace zero constraint, $\tilde{E}$ can be expressed as the linear combination of the three nullspace bases corresponding to the matrix of constraints:

$$\tilde{E} = xX + yY + zZ, \tag{22}$$

with $z = 1$ since E is homogeneous. The essential matrix has rank two and two equal non-zero singular values, such that:

$$\det(E) = 0, \tag{23}$$

$$2EE^\top E - \mathrm{tr}(EE^\top)E = 0. \tag{24}$$

Then, Equation 23 and Equation 24 give ten equations in terms of x and y with ten monomials: $\begin{bmatrix} x^3 & x^2y & x^2 & xy^2 & xy & x & y^3 & y^2 & y & 1 \end{bmatrix}$. By stacking the coefficients of the ten equations in a 10×10 matrix, the values of x and y can be computed from the vector corresponding to the smallest singular value of the matrix. Let v be the vector for the smallest singular value. Then $x = v_6/v_{10}$, $y = v_9/v_{10}$, and E is computed as in Equation 22. R and t are recovered from E as follows [13]. Let $E = U\,\mathrm{diag}(\sigma, \sigma, 0)V^\top$ be the SVD of the essential matrix, with U and V chosen such that $\det(U) > 0$ and $\det(V) > 0$. Then, R and t can be recovered from $t_u = u_{:,3}$ (the third column of U), $R_a = UDV^\top$, and $R_b = UD^\top V^\top$, where:

$$D = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{25}$$

with four different solutions: (R_a, t_u), $(R_a, -t_u)$, (R_b, t_u), and $(R_b, -t_u)$. Single solution selection is based on cheirality constraints (i.e. the observations are in front of the camera). Finally, note that $\mathrm{tr}(E) = 0$ and $\det(E) = 0$ imply that the eigenvalues of E have the form $[\lambda, -\lambda, 0]$. We found that enforcing this structure via eigendecomposition consistently yielded worse results, so we only enforce $[1, 1, 0]$ singular values for E for all three solvers.

3.5 NM6: Solver from six 2D-2D point matches

With six point correspondences and the trace zero constraint (Equation 21), $\tilde{E}$ can be expressed as the linear combination of the two nullspace bases corresponding to the matrix of constraints:

$$\tilde{E} = xX + yY, \tag{26}$$

with $y = 1$ since E is homogeneous. The ten equations given by Equation 23 and Equation 24 are now cubics in x, which are stacked in a 10×4 matrix. Let v be the vector corresponding to the smallest singular value of the equation matrix. Then $x = v_3/v_4$, and E is computed using Equation 26. Finally E is decomposed into R and t as described in the previous subsection.

3.6 NM7: Linear Solver from seven 2D-2D point matches

A linear solver, analogous to the eight point solver of Longuet-Higgings [18] for E, can be constructed from seven 2D-2D point correspondences and the trace zero constraint (Equation 21), by extracting the elements of E from the nullspace of the matrix of constraints. Finally E is decomposed into R and t.

4 Experiments

We compare the performance of our three hybrid solvers (hpc1, hpc2, and hpc3) and non-minimal solvers (nm5, nm6, and nm7) with the planar solvers of [9] (2p3d), and 4pst0 from [17]. We also compare with the solvers of [5] (choi), [12] (lsq), [6] (fast), which assume knowledge of the plane of motion. As at least three 2D-3D or five 2D-2D point correspondences are enough to estimate full DOF (non-planar) motion, we also compare with the full DOF solvers of [14] (horn), [22] (lambda), [20] (nister), and [18] (longuet). See Table 1 for a summary. All methods and experiments were implemented in MATLAB R2023a.

We adopt the experimental workflow of [9]. The solvers using only 2D-2D point correspondences cannot determine the scale of the camera translation due to the lack of 3D information. Therefore, to perform additional translation error comparisons we extend the 2D-2D solvers with a final step to compute the scale of the translation ρ from the 3D data available in the datasets:

$$\rho = \hat{t} \cdot (P_2 - RP_1), \tag{27}$$

where $\hat{t}$ corresponds to the direction of the translation and (P_1, P_2) are corresponding 3D points. An additional benefit of this extension is the sign disambiguation of t_u when decomposing the E into R and t.

4.1 Synthetic Data

For the synthetic data experiments we generated a dataset of 10000 random planar poses. The rotations are given about the Y axis and the translations are on the $X-Z$ plane, which allows us to draw comparisons w.r.t. the solvers that assume a fixed orientation of the plane of motion (i.e., [5, 6, 12]). In the case of multiple solutions (i.e., [6,12,17,20,22], hpc1, hpc2, and hpc3), we select the one closest (minimum error) to the ground truth orientation. We report the medians of the orientation error, the relative translation error, the plane of motion orientation (i.e., rotation axis) error, and the direction of translation error. The orientation error is computed as the angle of $R^\top R_{gt}$. The relative translation error is given by $\|t - t_{gt}\|/\|t_{gt}\|$. The plane of motion orientation error is the angle between k and k_{gt}. The direction of translation error is the angle between t and t_{gt}. The subscript $_{gt}$ denotes ground truth pose. Note that for the fixed orientation solvers (i.e., [5, 6, 12]), there is no plane of motion orientation error. The effect of the misalignment between the assumed and true plane orientation

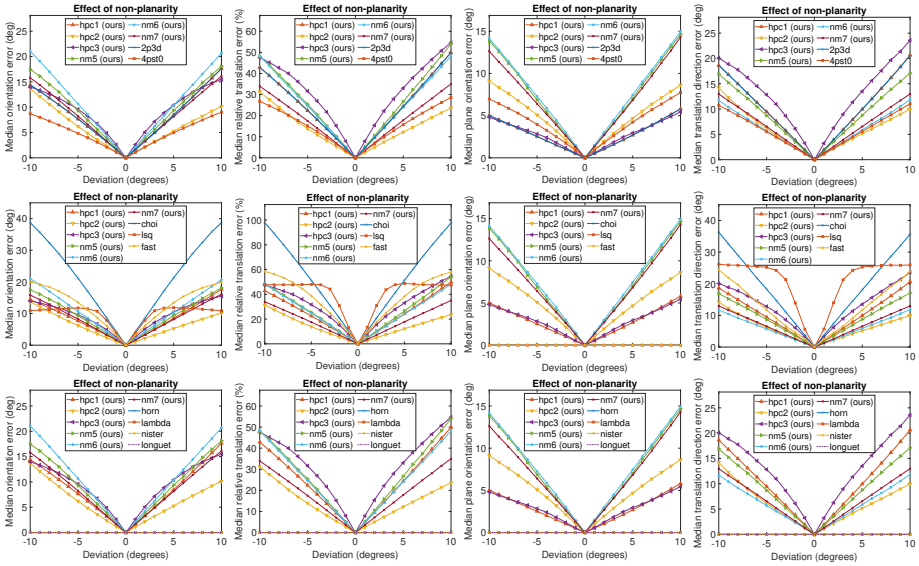

Fig. 2: Effect of non-planarity of motion. From top to bottom: comparison with other planar methods without plane of motion priors, comparison with solvers assuming a fixed plane of motion orientation, and comparison with full DOF motion solvers.

(i.e., ground plane calibration error) for the [5, 12] solvers has been explored in [9], showing some tolerance to small misalignments.

Validation. Although our dataset considers Y as the axis of rotation, we verified that our solvers can handle arbitrary plane-of-motion orientations by randomly rotating the poses in the dataset, i.e., $R' = R_g R R_g^\top$ and $t' = R_g t$, where R_g is a random rotation. This transformation preserves the planarity of motion while randomly reorienting the axis of rotation. The solvers leveraging epipolar constraints (i.e., hpc2, hpc3, nm5, nm6, and, nm7), cannot handle pure rotation since $E = 0$. However, pure translation is supported. As presented in section section 3, Equation 2 can also become 0 when $\alpha = n\pi$ for $n \in \mathbb{Z}$, corresponding to pure translation and 180 degrees rotations. We verified that our nm5, nm6, and nm7 solvers are able to handle both cases. Finally, we found that the nm5, nm6, and nm7 solvers fail when all of the input point correspondences are coplanar. Details can be found in the supplementary material.

Effect of non-planarity of motion. We evaluate performance of the solvers for various degrees of non-planarity. Non-planar motions are generated by fixing the rotation R and then rotating the vector t about the axis orthogonal to both the rotation axis and t. This transformation preserves the alignment of the rotation axis and the Y axis. The rotations of t are given in the range from -10 to 10 degrees in increments of 1 degree. The results are shown in Figure 2. Overall, our nm7 and hpc2 solvers achieve a performance in between [9] and [17], while hpc1 has the same non-planarity sensitivity as [9]. The hpc3, nm5, and nm6 solvers

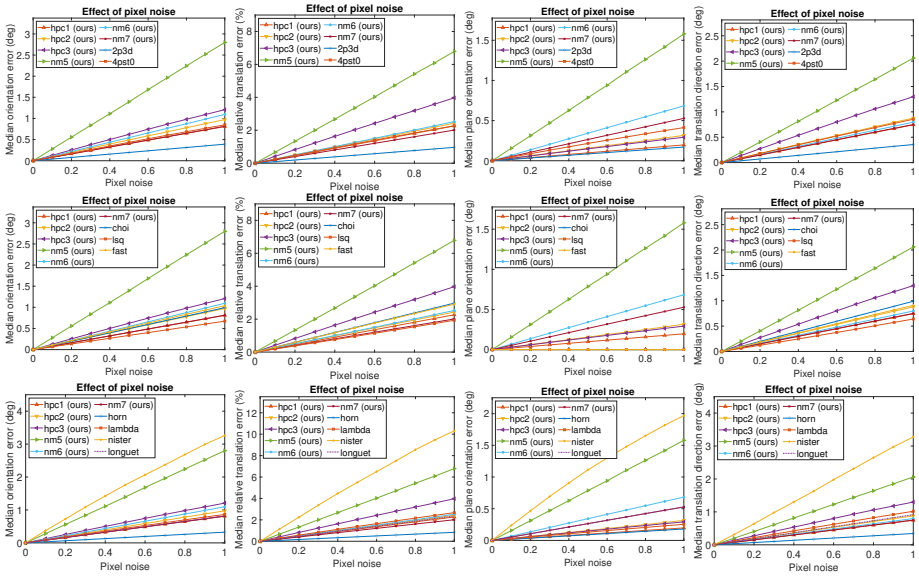

Fig. 3: Effect of 2D pixel noise. From top to bottom: comparison with other planar methods without plane of motion priors, comparison with solvers assuming a fixed plane of motion orientation, and comparison with full DOF motion solvers.

are the most sensitive. We observe that for non-planarity deviations less than 5 degrees, all of our solvers outperform the fixed-plane solvers [5, 6, 12]. The full DOF solvers (i.e., [14, 18, 20, 22]) clearly are not affected by non-planarities.

Effect of 2D pixel noise. We evaluate the effect of 2D measurement noise for increasing magnitude of noise. Pixel observations are disturbed by 2D Gaussian noise from 0 to 1 pixels in increments of 0.1 pixels, considering a focal length of $f = 720$ pixels. The corresponding depth values for the points are not modified, which is relevant for the solvers using 3D data (i.e., [5,9,14,22], hpc1, hpc2, and hpc3). The results are shown in Figure 3. All of our solvers are outperfomed by [9]. Compared to [17], the hpc1 and nm7 solvers are less sensitive, while hpc2 achieves a similar sensitivity, and the nm5, hpc3, nm6 solvers are more sensitive. Regarding the fixed orientation solvers, all of our solvers are more sensitive than [12]. Our hpc1, nm7, hpc2, and nm6 solvers are less sensitive than [5, 6], while the nm5 and hpc3 solvers are more sensitive. As for the full DOF solvers, all our solvers are less sensitive than [20] but they are all outperformed by [14]. Compared to [22], the nm5 and hpc3 solvers show increased sensitivity, the hpc2 and nm6 solvers yield a similar performance, and the hpc1, nm7 solvers are less sensitive. Finally, the hpc1 and nm7 solvers perform better than [18], while the hpc2 solver has a similar performance, and the hpc3, nm5, nm6 solvers are more sensitive. Overall, our nm5 and hpc3 solvers are the most sensitive.

Effect of depth noise. To evaluate the effect of depth measurement errors, pixel measurements are fixed and depth is adjusted by $uS\%$ of its magnitude

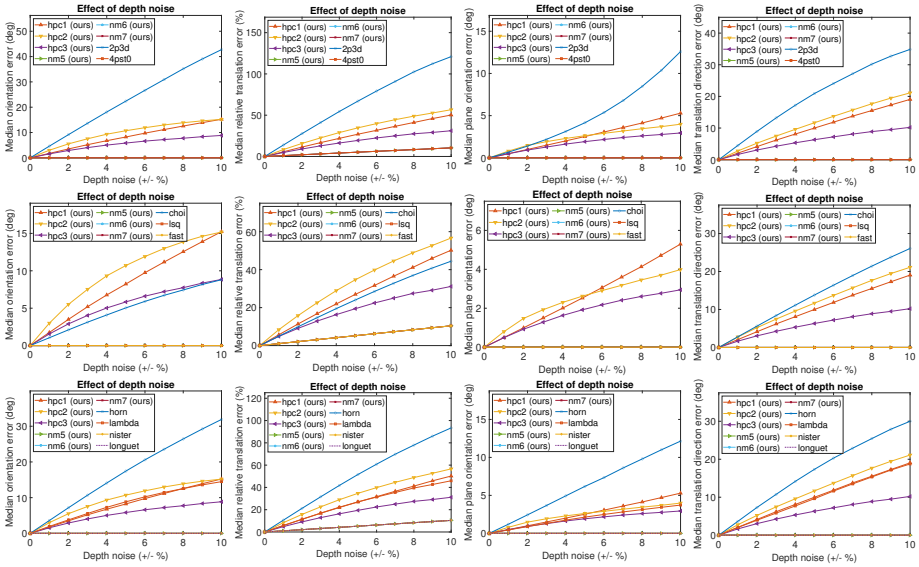

Fig. 4: Effect of depth noise. From top to bottom: comparison with other planar methods without plane of motion priors, comparison with solvers assuming a fixed plane of motion orientation, and comparison with full DOF motion solvers.

where $u \in [-1, 1]$ is a uniform random variable and S is the scale. For example, for $S = 5$ the new depth value will be somewhere between 95% and 105% of its original value. The results are shown in Figure 4 Of course, the solvers that use only 2D-2D point correspondences (i.e., [6, 12, 17, 18, 20], nm5, nm6, and nm7) are not affected (translations errors are due to the extension of Equation 27). We observe that the pure 3D-3D solvers (i.e., [9, 14]) are affected disproportionately compared to the rest of the solvers. The hpc2 and hpc1 solvers are more sensitive than [5], while hpc3 is less sensitive. Our hpc1 solver has a performance close to [22], while hpc3 is the least sensitive of the solvers using 3D data. Overall, solvers using at least one 3D-3D correspondence are more sensitive to depth noise than solvers using multiple 2D-3D correspondences.

Execution times. All solvers were implemented completely in MATLAB (no pre-compiled C/C++ MEX) and were run 30 times on the dataset for a total of 30×10000×2 samples per solver. Table 2 shows the median execution time for each solver. For methods returning multiple solutions we do not include the time for selecting one of the solutions. In general, we observe that solvers leveraging additional 3D data and/or priors yield simpler solutions achieving faster runtimes but with potentially more sensitivity to data/prior imperfections.

4.2 Real Data

We evaluate the performance of our solvers on the KITTI odometry dataset [10] and the TUM RGB-D dataset [24]. For KITTI, the stereo pair corresponding to

Table 2: Median execution time (μs) on the synthetic dataset (MATLAB).

Solver	μs	Solver	μs	Solver	μs	Solver	μs	Solver	μs
2p3d [9]	2.5	fast [6]	10.0	lambda [22]	23.0	longuet [18]	31.7	nister [20]	133.7
horn [14]	5.6	choi [5]	10.9	nm7 (ours)	24.8	nm6 (ours)	39.2	4pst0 [17]	155.7
hpc1 (ours)	8.7	lsq [12]	17.9	hpc2 (ours)	26.2	nm5 (ours)	53.9	hpc3 (ours)	262.3

the cameras 0 and 1 were used to extract 3D data via triangulation. For both datasets, SURF [1] features across sequential pairs of images were matched to obtain 3D-3D point correspondences. We integrate the solvers in a RANSAC framework with an adaptive number of iterations $N = \ln(1-p)/\ln(1-v^s)$, where $p = 0.9999$, v is the inlier ratio running estimate, and s is the number of correspondences required per solver. A reprojection error of 3 pixels on both images is used as the inlier threshold. For solvers with multiple solutions (i.e., [6, 12, 17, 20, 22], hpc1, hpc2, and hpc3), all solutions are tested on all points and the one with the largest support is selected. We report the medians of the orientation error (θ_R), the translation error ($\|t\|$), and the direction of translation error (θ_t), computed as the angle of $R^\top R_{gt}$, $\|t - t_{gt}\|$, and the angle between t and t_{gt}, respectively ($_{gt}$ denotes ground truth). We consider different stride values, i.e, pose is estimated for frame pairs n and $n+s$, where s is the stride.

Table 3: Median orientation error (in degrees), translation error (in meters), and direction of translation error (in degrees) for the first 11 sequences of the KITTI dataset [10] for strides 1, 2, and 5. For each category, best in bold and second best underlined.

	Stride 1			Stride 2			Stride 5		
	θ_R	$\|t\|$	θ_t	θ_R	$\|t\|$	θ_t	θ_R	$\|t\|$	θ_t
2p3d [9]	0.167	0.116	3.857	0.211	0.180	2.633	0.356	0.326	1.921
hpc1 (ours)	<u>0.128</u>	0.063	2.414	0.183	0.093	1.796	0.308	0.185	1.370
hpc2 (ours)	0.137	0.066	2.487	0.186	0.100	1.718	0.305	0.193	1.373
hpc3 (ours)	**0.110**	**0.046**	**2.016**	**0.156**	**0.075**	**1.497**	**0.259**	**0.162**	**1.225**
4pst0 [17]	0.130	0.066	2.331	<u>0.179</u>	0.093	1.744	0.299	0.199	1.333
nm5 (ours)	0.135	0.065	2.348	0.185	0.100	1.718	0.303	0.200	1.402
nm6 (ours)	0.129	<u>0.057</u>	<u>2.142</u>	0.181	<u>0.086</u>	<u>1.592</u>	<u>0.298</u>	<u>0.175</u>	<u>1.236</u>
nm7 (ours)	0.138	0.065	2.314	0.185	0.093	1.644	0.313	0.182	1.318
choi [5]	0.172	0.093	3.311	0.312	0.230	3.564	0.643	1.011	3.878
lsq [12]	0.181	0.099	3.507	0.308	0.224	3.249	0.581	1.019	3.270
fast [6]	0.185	0.116	3.604	0.318	0.286	3.649	0.594	1.130	3.459
horn [14]	0.211	0.116	4.957	0.256	0.172	3.205	0.398	0.346	2.706
lambda [22]	**0.080**	**0.030**	**1.311**	**0.103**	**0.048**	**1.042**	**0.173**	**0.118**	**0.899**
nister [20]	<u>0.106</u>	<u>0.056</u>	<u>2.016</u>	<u>0.124</u>	<u>0.078</u>	<u>1.435</u>	<u>0.187</u>	<u>0.158</u>	<u>1.046</u>
longuet [18]	0.243	0.126	3.982	0.249	0.132	1.971	0.278	0.201	1.263

KITTI dataset. For the KITTI dataset [10], we evaluate on the first 11 sequences for which ground truth data is available. The median results for all sequences are in Table 3. Our hpc3 and nm6 solvers outperform all of the other planar motion solvers. All of our solvers outperform the pure 3D-3D solvers [9, 14], the fixed orientation solvers [5, 6, 12], and achieve results close to [17, 18]. However, all of our solvers are outperformed by the full DOF methods of [20, 22].

Table 4: Median orientation error (in degrees), translation error (in meters), and direction of translation error (in degrees) for the sequences freiburg2 pioneer 360, slam, slam2, and slam3 of the TUM dataset [24] for strides 2, 4, and 6. For each category, best in bold and second best underlined.

	Stride 2			Stride 4			Stride 6		
	θ_R	$\|t\|$	θ_t	θ_R	$\|t\|$	θ_t	θ_R	$\|t\|$	θ_t
2p3d [9]	0.611	0.033	0.671	0.700	0.041	0.621	0.760	0.044	1.076
hpc1 (ours)	0.561	0.022	<u>0.184</u>	0.644	0.026	0.569	0.681	0.031	<u>0.946</u>
hpc2 (ours)	0.435	0.014	0.220	0.510	0.019	**0.330**	0.554	0.024	**0.851**
hpc3 (ours)	0.446	**0.012**	0.925	0.482	**0.016**	0.856	0.535	**0.020**	1.201
4pst0 [17]	0.434	<u>0.013</u>	**0.065**	<u>0.481</u>	<u>0.017</u>	<u>0.450</u>	0.542	<u>0.021</u>	0.947
nm5 (ours)	<u>0.424</u>	<u>0.013</u>	0.232	0.495	<u>0.017</u>	0.552	<u>0.532</u>	**0.020**	0.957
nm6 (ours)	**0.419**	**0.012**	0.610	**0.475**	**0.016**	0.702	**0.526**	**0.020**	1.130
nm7 (ours)	0.425	<u>0.013</u>	0.810	0.490	<u>0.017</u>	0.938	0.536	<u>0.021</u>	1.252
choi [5]	0.626	0.042	1.843	0.858	0.076	2.579	1.089	0.107	2.983
lsq [12]	0.483	0.024	3.463	0.664	0.045	3.884	0.848	0.066	3.922
fast [6]	0.490	0.025	3.294	0.672	0.045	3.703	0.844	0.068	3.858
horn [14]	0.738	0.043	1.372	0.837	0.053	**1.587**	0.930	0.061	**1.671**
lambda [22]	0.530	<u>0.017</u>	1.552	0.587	0.022	1.875	0.642	0.028	1.874
nister [20]	**0.445**	**0.014**	<u>1.116</u>	**0.503**	**0.018**	<u>1.639</u>	**0.544**	**0.023**	<u>1.749</u>
longuet [18]	<u>0.450</u>	**0.014**	**1.104**	<u>0.520</u>	<u>0.019</u>	1.715	<u>0.573</u>	<u>0.024</u>	1.757

TUM dataset. For the TUM RGBD dataset [24], we evaluate on the four sequences freiburg2 pioneer 360, freiburg2 pioneer slam, freiburg2 pioneer slam2, and freiburg2 pioneer slam3. The median results for all sequences are presented in Table 4. Our nm6 solver outperforms all of the solvers. Again, all of our solvers outperform the pure 3D-3D solvers [9,14] and the fixed orientation solvers [5, 6, 12]. Our hpc2, hpc3, nm5, and nm7 solvers outperform all of the full DOF solvers and [9], while achieving results close to [17].

Execution Times. The median number of RANSAC iterations and total execution time for each solver on the real datasets is shown in Table 5. Strikingly, we observe that hpc1 achieves better performance than [9] with essentially the same runtime on both datasets, just by dropping the 3D data of one of the input points and enforcing length consistency between views (Equation 5). All of our solvers except hpc3 (which is the slowest of all) runs faster than [17], with nm6 achieving better performance (and hpc2, hpc3, nm5, nm6, nm7 with similar performance) on both datasets. Overall, the 3D-3D solvers [9, 14] run faster than

Table 5: Median execution time (ms) on the real datasets. Left: KITTI. Right: TUM.

Solver	Iterations	Time	Solver	Iterations	Time
hpc1 (ours)	9	1.1	2p3d [9]	11	0.8
2p3d [9]	19	1.1	hpc1 (ours)	8	0.9
lambda [22]	7	1.5	horn [14]	21	1.0
choi [5]	16	1.7	choi [5]	12	1.4
longuet [18]	17	2.1	lambda [22]	8	1.5
horn [14]	55	2.6	nm5 (ours)	12	1.6
hpc2 (ours)	14	3.4	nm6 (ours)	13	1.8
fast [6]	25	3.9	hpc2 (ours)	8	2.0
nm6 (ours)	29	4.1	longuet [18]	18	2.0
nm5 (ours)	26	4.3	nm7 (ours)	17	2.1
nm7 (ours)	38	4.7	fast [6]	12	2.2
lsq [12]	32	8.7	lsq [12]	15	5.7
nister [20]	20	12.8	nister [20]	10	6.4
4pst0 [17]	18	18.2	4pst0 [17]	9	8.1
hpc3 (ours)	9	31.1	hpc3 (ours)	6	13.5

our solvers but with higher estimation errors. Compared to the fixed orientation solvers [5, 6, 12], our solvers achieve better performance but [5] is consistently faster (and [6] faster than nm5, nm6, nm7 on KITTI) and [12] is consistently slower on both datasets. Our solvers (except hpc3) are faster than [20] but slower than [22] (except hpc1). Moreover, [20,22] achieve better performance on KITTI but not on TUM. Finally, [18] is faster on KITTI and slower on TUM, with overall worse results than our solvers. We attribute the performance difference across the datasets to the higher non-planarities present in the KITTI dataset.

5 Conclusion

We introduced three new minimal solvers from hybrid point correspondences and three non-minimal solvers from pure 2D-2D point correspondences, enabling various tradeoffs between input cardinality, stability, accuracy, and run time in the context of planar motion. These solvers are general in the sense that they do not require prior knowledge about the motion plane, and adapt to variations in the disposition between the camera and the plane of motion without the need for additional sensors and/or priors. Our experimental evaluation demonstrated that our solvers can yield better estimates (with comparable runtimes) than pure general and fixed orientation solvers, and in some cases (TUM) even outperform full DOF solvers. As for limitations, our hpc2, hpc3, nm5, nm6, and nm7 solvers are unstable under pure rotation as they rely on epipolar constraints. Nevertheless, pure translation is acceptable. Our nm5, nm6, and nm7 cannot handle all-coplanar inputs. The code is available at `https://github.com/jdibenes/gpm`.
Acknowledgement. Work sponsored by Defense Advanced Research Projects Agency contract HR00112220003, contents do not necessarily reflect the position of the U.S. Government, and no official endorsement should be inferred.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) Computer Vision – ECCV 2006. pp. 404–417. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
2. Bhayani, S., Sattler, T., Larsson, V., Heikkilä, J., Kukelova, Z.: Partially calibrated semi-generalized pose from hybrid point correspondences. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2881–2890 (2023). https://doi.org/10.1109/WACV56688.2023.00290
3. Booij, O., Zivkovic, Z.: The planar two point algorithm. Geojournal (2009)
4. Camposeco, F., Cohen, A., Pollefeys, M., Sattler, T.: Hybrid camera pose estimation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 136–144 (2018). https://doi.org/10.1109/CVPR.2018.00022
5. Choi, S.I., Park, S.Y.: A new 2-point absolute pose estimation algorithm under planar motion. Advanced Robotics **29**, 1–9 (05 2015). https://doi.org/10.1080/01691864.2015.1024285
6. Choi, S., Kim, J.H.: Fast and reliable minimal relative pose estimation under planar motion. Image and Vision Computing **69**, 103–112 (2018). https://doi.org/https://doi.org/10.1016/j.imavis.2017.08.007, https://www.sciencedirect.com/science/article/pii/S0262885617301233
7. Choi, S., Park, J., Yu, W.: Simplified epipolar geometry for real-time monocular visual odometry on roads. International Journal of Control, Automation and Systems **13** (07 2015). https://doi.org/10.1007/s12555-014-0157-6
8. Chou, C.C., Wang, C.C.: 2-point ransac for scene image matching under large viewpoint changes. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 3646–3651 (2015). https://doi.org/10.1109/ICRA.2015.7139705
9. Dibene, J.C., Min, Z., Dunn, E.: General planar motion from a pair of 3d correspondences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8060–8070 (October 2023)
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
11. Goedeme, T., Tuytelaars, T., Van Gool, L., Vanacker, G., Nuttin, M.: Feature based omnidirectional sparse visual path following. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1806–1811 (2005). https://doi.org/10.1109/IROS.2005.1545111
12. Hajder, L., Barath, D.: Least-squares optimal relative planar motion for vehicle-mounted cameras. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 8644–8650 (2020). https://doi.org/10.1109/ICRA40945.2020.9196755
13. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edn. (2004)
14. Horn, B.: Robot Vision. MIT electrical engineering and computer science series, McGraw-Hill Book Company (1986), https://books.google.com/books?id=jpX9Lrxn58MC
15. Josephson, K., Byrod, M., Kahl, F., Astrom, K.: Image-based localization using hybrid feature correspondences. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2007). https://doi.org/10.1109/CVPR.2007.383353

16. Li, B., Larsson, V.: Gaps: Generator for automatic polynomial solvers. arXiv preprint arXiv:2004.11765 (2020)
17. Li, B., Martyushev, E., Lee, G.H.: Relative pose estimation of calibrated cameras with known se(3) invariants. In: Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX. p. 215–231. Springer-Verlag, Berlin, Heidelberg (2020)
18. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. Nature **293**, 133–135 (Sep 1981). https://doi.org/10.1038/293133a0
19. Maybank, S.: Theory of Reconstruction from Image Motion. Springer Berlin Heidelberg (1993). https://doi.org/10.1007/978-3-642-77557-4
20. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**, 756–770 (2004), https://api.semanticscholar.org/CorpusID:886598
21. Ortín, D., Montiel, J.M.M.: Indoor robot motion based on monocular images. Robotica **19**(3), 331–342 (may 2001). https://doi.org/10.1017/S0263574700003143, https://doi.org/10.1017/S0263574700003143
22. Persson, M., Nordberg, K.: Lambda twist: An accurate fast robust perspective three point (p3p) solver. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
23. Philip, J.: A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. The Photogrammetric Record **15**(88), 589–599 (1996). https://doi.org/https://doi.org/10.1111/0031-868X.00066, https://onlinelibrary.wiley.com/doi/abs/10.1111/0031-868X.00066
24. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the International Conference on Intelligent Robot Systems (IROS) (Oct 2012)
25. Troiani, C., Martinelli, A., Laugier, C., Scaramuzza, D.: 1-point-based monocular motion estimation for computationally-limited micro aerial vehicles. In: 2013 European Conference on Mobile Robots. pp. 13–18 (2013). https://doi.org/10.1109/ECMR.2013.6698813
26. Troiani, C., Martinelli, A., Laugier, C., Scaramuzza, D.: 2-point-based outlier rejection for camera-imu systems with applications to micro aerial vehicles. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 5530–5536 (2014). https://doi.org/10.1109/ICRA.2014.6907672
27. Zhang, Y., Liang, W., Li, Y., An, H., Tan, J.: Orientation estimation using visual and inertial sensors. In: 2015 IEEE International Conference on Information and Automation. pp. 1871–1876 (2015). https://doi.org/10.1109/ICInfA.2015.7279593
28. Zhao, J., Kneip, L., He, Y., Ma, J.: Minimal case relative pose computation using ray-point-ray features. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(5), 1176–1190 (2020). https://doi.org/10.1109/TPAMI.2019.2892372