General Planar Motion from a Pair of 3D Correspondences

Juan Carlos Dibene Zhixiang Min Enrique Dunn
Stevens Institute of Technology
{jdibenes,zmin1,edunn}@stevens.edu

Abstract

We present a novel 2-point method for estimating the relative pose of a camera undergoing planar motion from 3D data (e.g. from a calibrated stereo setup or an RGB-D sensor). Unlike prior art, our formulation does not assume knowledge of the plane of motion, (e.g. parallelism between the optical axis and motion plane) to resolve the under-constrained nature of SE(3) motion estimation in this context. Instead, we enforce geometric constraints identifying, in closed-form, a unique planar motion solution from an orbital set of geometrically consistent SE(3) motion estimates. We explore the set of special and degenerate geometric cases arising from our formulation. Experiments on synthetic data characterize the sensitivity of our estimation framework to measurement noise and different types of observed motion. We integrate our solver within a RANSAC framework and demonstrate robust operation on standard benchmark sequences of real-world imagery. Code is available at: `https://github.com/jdibenes/gpm`.

1. Introduction

Pose estimation is a basic geometric perception task crucial to multiple computer vision and robotic applications such as 3D modeling [9], image registration [22], object pose estimation [24], visual odometry [23], and simultaneous localization and mapping (SLAM) [21]. The problem is generally formulated as the estimation of the parameters of a motion-model representation (i.e. SE(2) or SE(3)) that satisfies a given image formation model and a set of input observations [8]. From a geometric perspective, the 2D and 3D variants of the absolute orientation problem (i.e. finding the rigid motion among two sets of corresponding points expressed in different Euclidean coordinate systems), are perhaps the most straightforward and well-known.

The use of constrained planar motion models allows to reduce data requirements and enforce structured geometric priors onto the estimates (e.g. non-holonomic constraints applicable to robotic vehicle platforms). For such cases, the desired estimate may be geometrically represented as an element of SE(2) defined on a 2D plane embedded in Euclidean 3-space. Surprisingly, the task of jointly estimating a completely unknown embedding motion plane, as well as the corresponding SE(2) motion parameters, remains an open research problem.

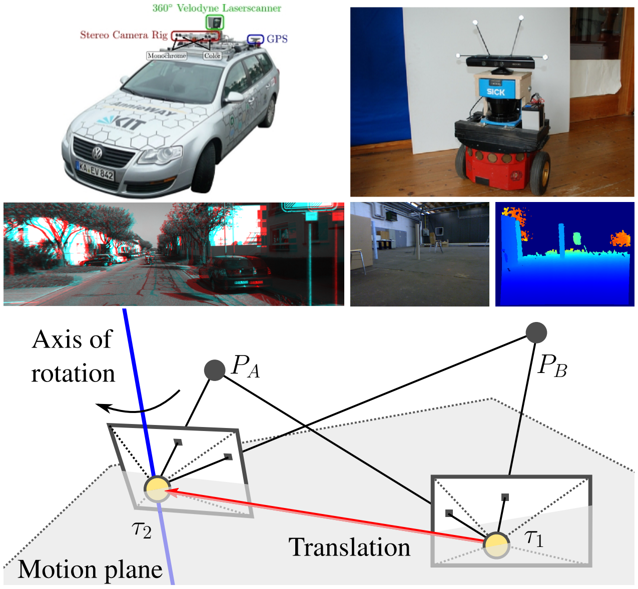

Figure 1. Planar motion is common to mobile platforms traversing locally flat terrain such as the KITTI [10] (left) and TUM [34] (right) benchmarks. Our method operates on 3D feature inputs, possibly attained through stereo triangulation or depth sensors, to estimate planar motion without any priors on the motion plane.

Our geometric formulation assumes the availability of a pair of correspondences among local (i.e. camera-space) 3D points attained from RGB-D inputs (e.g. active/passive stereo, time-of-flight sensors, view-registered lidar). Per the discussion in Horn's classic work [16], such input under-constrains the estimation of the general SE(3) transformation relating the 3D coordinates. Our formulation disambiguates among the ensuing 1D family of solutions in general SE(3) by enforcing compliance to a general planar motion (i.e. SE(2)) model through a straightforward geometric constraint: orthogonality between the translation component and the rotation axis.

Our framework defines the geometry of the set of possible SE(3) solutions, and explicitly models the 1D parametric dependency for both general motion components (i.e. 3D rotation and translation) to derive a closed-form solution yielding the unique SE(3) element satisfying the general planar motion assumption. The technical contributions and differentiators of our work are:

1. **Simplicity.** We develop a minimal and purely geometric first-order analytic solution defined over linear vector operations and trigonometric functions instead of root-searching over higher-order polynomial terms.

2. **Generality.** Our formulation makes no assumptions on the camera disposition w.r.t. the *ground* motion plane, making it suitable for deployment in capture rigs with unknown (but constant) configuration.

3. **Interpretability.** By virtue of its explicit geometric parameterization, our formulation can summarily identify and robustly estimate special motion instances such as pure rotation and/or translation.

2. Related work

We present a summary of closely related work for relative pose estimation assuming planar motion.

Ortín and Montiel [27] use monocular vision to estimate the relative motion of an indoor robot limited to the X-Z plane. They propose a linear 3-point algorithm, and an iterative non-linear 2-point solver based on a fundamental matrix parametrization. For the case of pure rotation, where the epipolar geometry is not defined, they consider an homography-based model from a 1-point match.

Stewénius and Åström [33] consider a multi-camera system undergoing planar motion and address 3 particular cases: 1) 8 points in 3 one-dimensional views, 2) 3 points in 2 two-dimensional views, and 3) 2 points in 3 two-dimensional views.

Booij and Zivkovic [2] provide a comprehensive description of the monocular planar 2-point algorithm of Goedemé et al. [11], for which the motion is constrained to the X-Y plane. The method determines relative heading and rotation angles, and can yield up to 2 possible solutions for the relative translation and rotation.

Choi et al. [4] introduce a 2-point solver for planar motion and a 1-point solver for circular planar motion. Both motions are restricted to the X-Z plane. The relative pose can be estimated from either a reduced essential matrix model with 4 solutions for the planar motion and 2 solutions for the circular motion, or from a reduced homography of the ground plane, with 1 solution for the planar case and 2 solutions for the circular case.

Nicolás et al. [20] address the problem of visual control for mobile robots with non-holonomic constraints. The robot undergoes planar motion and only the x-coordinates of the epipoles are considered for control design.

Scaramuzza [32] presents a method to estimate the relative motion of a vehicle from 1 point correspondence. The assumptions are that the vehicle motion follows the Ackermann steering principle, that motion is on the X-Y plane, and that the distance between the camera and the back wheel axis is zero.

Lee at al. [18] present a minimal 2-point solver for the relative motion of a generalized camera under the Ackermann motion model, with up to 6 solutions. Motion is considered on the X-Y plane.

Miraldo and Araujo [25] address relative pose estimation under planar motion considering known 3D lines. It does not require to determine correspondences between pixels and 3D points. Instead, what it is required is to determine for each pixel to which 3D line it belongs to. The relative motion is constrained to the X-Y plane.

Choi et al. [5] present a relaxed planar motion model which allows for small non-planar motion. The planar component is assumed to lie on the X-Z plane and the relative pose is estimated from 3 point correspondences.

Chou and Wang [7] consider planar motion on the X-Y plane. From the epipolar constraint, they derive a system of equations that can be interpreted as the intersection of 2 ellipsoids, which yields up to four solutions. Their solver requires 2 point correspondences.

Choi and Kim [3] build upon the derivation of [7] to formulate the planar relative pose problem as the intersection of a line and a unit circle. Their 2-point solver yields up to two solutions. Motion is on the X-Z plane.

Saurer et al. [31] present minimal solutions for the egomotion of a camera based on homography knowing the gravity direction between calibrated images. The solutions depend on the prior knowledge about the reference plane used by the homography.

Örnhag and Heyden [26] consider two cameras on a mobile platform directed towards the floor and mounted at the same distance from the ground. Planar motion is assumed (the mobile platform rotates about the z-axis) and the relative motion is estimated from ground plane homographies.

Hajder and Barath [13] present a least squares solver for relative planar motion from 3 point correspondences, considering motion on the X-Z plane. The solver obtains the motion parameters as the roots of a 6th degree polynomial.

Zhang et al. [35] propose a 2-point solver for relative planar motion, on the X-Z plane, from ground plane homographies. Their solver computes the rotation and translation independently so it can work with pure rotation scenes.

Zhao et al. [36] solve for relative motion using ray-point-ray correspondences. In the case of planar motion, the rela-

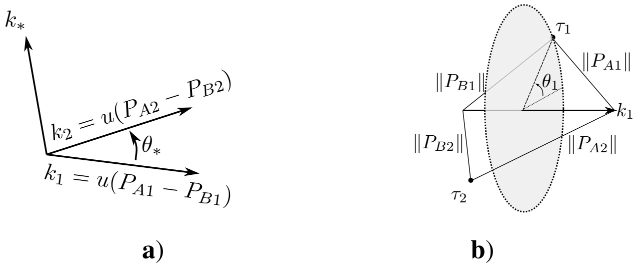

Figure 2. Our method factorizes the rotation component R of the motion into a product two rotations $R = R_* R_1$. **a**) The particular solution R_*, a rotation about k_*, aligns the vectors given by the 3D point pairs. **b**) Combining R_* with R_1, a rotation about k_1, yields the general solution to R parameterized by θ_1, whose value is determined from the planar motion constraint.

tive pose can be estimated from 1 ray-point-ray correspondence. For the planar case, motion is on the X-Y plane.

Li et al. [19] present a comprehensive study of relative pose estimation constrained by $SE(3)$ invariants, including planar motion. They show that rotation angle and screw translation can be integrated into relative pose solvers to reduce the number of minimal points. As these are $SE(3)$ invariants, extrinsic pose of the camera with respect to the motion plane is not required.

Hajder and Barath [14] propose 2 solvers to estimate the extrinsic camera parameters from a single affine correspondence assuming general planar motion. The image plane is assume to be orthogonal to the ground (X-Z plane). The first solver corresponds to the calibrated camera case, while the seconds considers an unknown common focal length.

Guan et al. [12] address the minimal cases for estimating the generalized relative pose from affine correspondences. For the case of planar motion, they present a minimal solver from 1 affine correspondence and a solver from 2 affine correspondences to overcome the degenerate case. Motion is on the X-Z plane.

Most related to our work, is the 2-point solver of Choi and Park [6] from 2D-3D point correspondences obtained from RGB-D data. However, they assume the motion of the camera is on the X-Z plane, which requires that the RGB-D camera is adjusted precisely to make sure the X-Z plane of the camera coordinate system is parallel to the moving plane of the robot.

3. General Planar Motion from a 3D Point Pair

It is common practice to consider SE(2) transformations to be applied to elements residing in Euclidean 2-space (i.e. contained within a plane). Our notion of general planar motions (GPMs) refers to those elements of SE(3) (applicable to elements of Euclidean 3-space) which are compatible with an SE(2) rigid motion. The geometric property defining such compatibility is having the rotation axis and the translation components represented by orthogonal vectors. Such notion defines GPMs as a subset of SE(3) and offers generalizations to higher dimensions (which are considered out of scope for this work). Geometrically, GPMs implicitly define the embedded motion plane by 1) specifying the plane's normal vector orientation to be parallel to the 3D axis of their rotation component, and 2) specifying the translational component vector to correspond to a point belonging to such plane.

The empirical relevance of such formulation is that it obviates the need to calibrate the camera w.r.t. the ground plane. As discussed in section 2, pose estimation can be significantly simplified by modeling constraints between the geometry of the image plane and the motion plane. While such works may be applicable to capture setups with arbitrary camera orientation (w.r.t. to the motion plane) by applying pre-processing steps to the input data, such arbitrary orientation must be known *a-priori*. Our 2-point GPM solver, requires no such information.

3.1. Geometric Solution from a Single 3D Point Pair

Our method considers two static 3D points, P_A and P_B, observed by a camera at an initial location τ_1 and then at a different location τ_2 (see Figure 1). The 3D points at τ_1, P_{A1} and P_{B1}, and those at τ_2, P_{A2} and P_{B2}, are given w.r.t. each camera's coordinate frame. Then, the points observed at τ_1 and τ_2 are related by a rotation and a translation as

$$P_{A2} = RP_{A1} + t, \tag{1}$$

$$P_{B2} = RP_{B1} + t, \tag{2}$$

where R and t are the parameters to be estimated. For planar motion, this problem has 5 degrees of freedom (DOF): 2 for the axis of rotation, 1 for the angle of rotation, 1 for the direction of the translation (on the plane of motion) and 1 for the magnitude of the translation. Equation 1 and Equation 2 provide 5 constraints on the 6-DOF pose (R, t) (since the distance between the points is invariant), with the remaining DOF corresponding to a rotation about the axis passing through both points. This last DOF is constrained by the planar motion assumption.

Factorization of the rotation. Our solution is based on factorizing R into two separate rotations. Similar approaches have been used in previous works such as [30, 29, 17]. Subtracting Equation 2 from Equation 1 yields

$$P_{A2} - P_{B2} = R(P_{A1} - P_{B1}). \tag{3}$$

Such an expression simply defines the rotation of the 3D line-segment vector across time. A particular solution R_* for Equation 3 can be obtained as follows. First, determine the axis k_* of the rotation between the points at τ_2 and those at τ_1, as the common normal to the line-segment vectors (see Figure 2**a**) by

$$k_* = u((P_{A1} - P_{B1}) \times (P_{A2} - P_{B2})), \tag{4}$$

where $u(\cdot)$ converts to a unit vector. For now, we assume that the cross product in Equation 4 is non-zero. The alternative case will be addressed in subsection 3.2. Next, the angle of rotation θ_* is

$$\theta_* = \arccos{(k_1 \cdot k_2)}, \tag{5}$$

where

$$k_1 = u(P_{A1} - P_{B1}), \tag{6}$$

$$k_2 = u(P_{A2} - P_{B2}). \tag{7}$$

Finally, Rodrigues' rotation formula gives

$$R_* = C(k_*, \theta_*) = I + \sin{(\theta_*)} K_* + (1 - \cos{(\theta_*)} K_*^2), \tag{8}$$

where K_* is the skew-symmetric matrix corresponding to k_*. We use $C(k, \theta)$ to denote the rotation matrix corresponding to a rotation about an axis k by an angle θ. Of course, R_* is only one of the infinitely many solutions for Equation 3. Geometrically, the entire family of rotations can be generated by combining R_* with a rotation along an axis parallel to either k_1 or k_2 (see Figure 2**b**). Accordingly, the remaining solutions can be generated by (and parameterized w.r.t.) a single parameter θ_1 as

$$P_{A2} - P_{B2} = R_* C\left(k_1, \theta_1\right)\left(P_{A1} - P_{B1}\right), \tag{9}$$

This parameterization is based on the fact that points lying on the axis of rotation remain fixed. The reader might wonder why is this parameterization sufficient without also considering rotations about the axis defined by P_{A2} and P_{B2}

$$P_{A2} - P_{B2} = C\left(k_2, \theta_2\right) R_* (P_{A1} - P_{B1}). \tag{10}$$

or even

$$P_{A2} - P_{B2} = C\left(k_2, \theta_2\right) R_* C\left(k_1, \theta_1\right)\left(P_{A1} - P_{B1}\right). \tag{11}$$

It can be shown that the parameterizations in Equation 9, Equation 10, and Equation 11 are all equivalent by using a known property of rotation matrices

$$R_* C(k_1, \theta) R_*^\top = C(R_* k_1, \theta) = C(k_2, \theta). \tag{12}$$

As a consequence, the same final composite rotation matrix R may be generated by either of our potential parameterizations. Thus, for simplicity, we chose the single parameter model defined by Equation 9

$$R = C(k_R, \theta_R) = C(k_*, \theta_*) C(k_1, \theta_1), \tag{13}$$

where k_R and θ_R are the axis and the angle corresponding to R, respectively.

Enforcing the GPM constraint. We use the planar motion constraint to determine the value of θ_1 as follows. The planar motion constraint is

$$(-R^\top t) \cdot k_R = 0. \tag{14}$$

Combining Equation 1 with Equation 14 and using the property of rotation matrices $R k_R = R^\top k_R = k_R$ along with the invariance of the dot product under rotations yields

$$\Delta_P \cdot k_R = 0, \tag{15}$$

where

$$\Delta_P = P_{A1} - P_{A2}. \tag{16}$$

Note that a similar result can be achieved using Equation 2. We assume for now that $\Delta_P \neq 0$. The opposite case is discussed in subsection 3.2.

Solving for the free parameter. From the Rodrigues formula for the composition of two axis-angle rotations

$$\begin{aligned} \sin\left(\frac{\theta_R}{2}\right) k_R = {} & \sin\left(\frac{\theta_*}{2}\right) \cos\left(\frac{\theta_1}{2}\right) k_* + \\ & \cos\left(\frac{\theta_*}{2}\right) \sin\left(\frac{\theta_1}{2}\right) k_1 + \\ & \sin\left(\frac{\theta_*}{2}\right) \sin\left(\frac{\theta_1}{2}\right) (k_* \times k_1). \end{aligned} \tag{17}$$

Finally, combining Equation 15 and Equation 17 yields

$$\theta_1 = 2 \arctan\left(\frac{y}{x}\right), \tag{18}$$

where

$$y = -\left(\sin\left(\frac{\theta_*}{2}\right) k_*\right) \cdot \Delta_P, \tag{19}$$

$$x = \left(\cos\left(\frac{\theta_*}{2}\right) k_1 + \sin\left(\frac{\theta_*}{2}\right) (k_* \times k_1)\right) \cdot \Delta_P. \tag{20}$$

Solving for the motion and the plane. The value of R is obtained by plugging θ_1 into Equation 13. Then t can be obtained from either Equation 1 or Equation 2

$$t = P_{A2} - R P_{A1}. \tag{21}$$

For estimating the plane of motion, the plane normal (k_R) can be obtained from Equation 17 given that $R \neq I$, where I is the 3×3 identity matrix. Note that the plane of motion always passes through the origin.

3.2. Geometric Special Cases

When the cross product yielding the axis k_* in Equation 4 is zero, it indicates the 3D vectors connecting the 3D points observed at τ_1 and τ_2 are parallel. There are two different geometric interpretations for this, each corresponding to a special case for our method: $\theta_* = 0$ or $\theta_* = \pi$. We assume $\theta = [0, \pi]$ since the direction of rotation is given by the axis k, following the right hand rule.

If $\theta_* = \pi$, then any axis k_* orthogonal to k_1 (or k_2) suffices. Such axis can be obtained, for example, from the SVD of $k_1^\top$ (or $k_2^\top$). Then, R and t can be estimated as described in section 3.

If $\theta_* = 0$, then the geometric configuration is degenerate. The rotation axis k_R is parallel to both k_1 and k_2 (see Figure 3**a**) and the planar motion constraint (Equation 15) is satisfied for all values of θ_1. Otherwise, the motion would violate the planar assumption. Therefore, it is not possible to recover the true value of θ_1. Note that both the numerator and denominator of Equation 18, which gives the solution for θ_1, vanish in this case and the result is undefined. Since $\theta_* = 0$ and $\theta_1 = 0$ yield $R = I$, pure translation and the degenerate case are indistinguishable. For real images, with potentially many more than two point correspondences (e.g. running in a RANSAC framework), the degenerate case with $\theta_1 \neq 0$ is seldom observed since all of the observed 3D points would have to be collinear and their line parallel to the axis of rotation. Therefore, our solver defaults to pure translation under the degenerate condition.

When the value of Δ_P in Equation 15 is zero both the numerator and denominator of Equation 18 vanish. We show the conditions under which this situation occurs. Having $\Delta_P = 0$ means that $P_{A2} = P_{A1}$. Thus, from Equation 1

$$P_{A1} = RP_{A1} + t. \tag{22}$$

Then, we eliminate R and solve for $\|t\|$

$$\|t\| = 2\|P_{A1}\| cos(\theta_{tP_{A1}}), \tag{23}$$

where $\theta_{tP_{A1}}$ is the angle between P_{A1} and t. Then, t can be parameterized by $\theta_{tP_{A1}}$ and $\theta_{P_{A1}}$ as

$$t = 2\cos(\theta_{tP_{A1}}) C(C(\hat{P}_{A1}, \theta_{P_{A1}}) k_\perp, \theta_{tP_{A1}}) P_{A1}, \tag{24}$$

where $\hat{P}_{A1} = u(P_{A1})$ and $k_\perp$ is any axis orthogonal to P_{A1}. Note that the parameter $\theta_{P_{A1}}$ encodes all possible choices of $k_\perp$. Equation 24 describes a sphere surface, centered on P_{A1} and with radius $\|P_{A1}\|$, for all t that satisfy $\Delta_P = 0$ (see Figure 3**b**). Then, from Equation 1, every t on the surface of the sphere has a corresponding family of rotations parameterized by θ_f

$$R = C(C(\hat{P}_{A1}, \theta_{P_{A1}}) k_\perp, 2\theta_{tP_{A1}} - \pi) C(\hat{P}_{A1}, \theta_f). \tag{25}$$

As an example, we show the two extreme cases. For $t = 0$ there is a family of $R = C(\hat{P}_{A1}, \theta_f)$, which corresponds to all pure rotations about P_{A1}. For $t = 2P_{A1}$ there is a family of $R = C(C(\hat{P}_{A1}, \theta_{P_{A1}}) k_\perp, \pi) C(\hat{P}_{A1}, \theta_f)$. In real images (with many point correspondences and a RANSAC framework), all of these special structures can be avoided just by choosing another set of point correspondences with $\Delta_P \neq 0$. The case where all samples fall in this special case is rarely observed, since it requires that t is on all of the spheres corresponding to all observed 3D points and that R belongs to all families given by all (t, sphere) pairs. For example, $t = 0$ is on all of the spheres and requires that the axis of rotation passes through all of the 3D points. Finally, similar results can be obtained from Equation 2 instead when considering $\Delta = P_{B1} - P_{B2}$ in section 3.

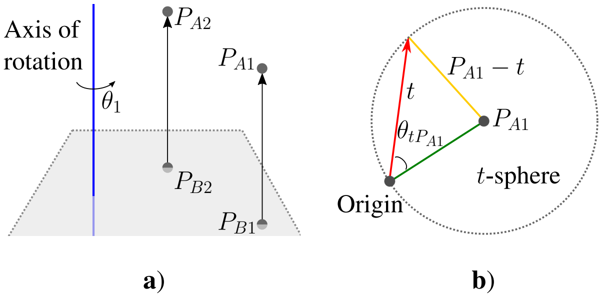

Figure 3. Geometric special cases. **a**) When $\theta_* = 0$ there is always a corresponding t for any θ_1, so it is not possible to disambiguate by the planar motion constraint. **b**) For $\Delta_P = 0$, it is necessary for t to be on the sphere centered on P_{A1} and for R to belong to a particular set. The planar motion constraint is always zero and cannot be used to disambiguate.

4. Experiments

We evaluate the performance of our solver and compare with state of the art methods on synthetic and real data. All of the experiments were run on MATLAB R2020b. We compare with the methods of Scaramuzza [32] (1pt-1*dof-2d2d), Choi and Park [6] (2pt-3dof-2d3d), Horn [15] (3pt-6dof-3d3d), Persson and Nordberg [28] (3pt-6dof-2d3d), and Hajder and Barath [13] (3pt-2*dof-2d2d). We denote all methods using a three-part name, where the first part is the number of input point correspondences, the second part is the DOF of the pose, and the third part is the type of point correspondences. For example, our method takes two 3D-3D point correspondences to estimate a 5-DOF pose and thus is denoted as 2pt-5dof-3d3d.

We use a pair of 6-DOF methods [15, 28] as an integrity baseline since they are not affected by deviations from planar motions. The 1pt-1*dof-2d2d method [32] allows to compare our method under the Ackermann motion model on a known motion plane. The remaining planar method pair [6, 13] require prior calibration of the ground plane and allow us to draw comparisons w.r.t. general planar motion model. Also, the different types of point correspondences

(2D-2D, 2D-3D) allow to better compare the performance and robustness of our solver under the effect of 2D and 3D noise. Note that the 3pt-2*dof-2d2d method [13] returns up to 6 solutions and 3pt-6dof-2d3d [28] returns up to 4.

The 2D-2D methods do not estimate the scale of the translation so we extended them to obtain the scale from the 3D data to have a uniform metric for the translation error. From Equation 1, the scale ρ of the translation is

$$\rho = \hat{t} \cdot (P_{A2} - RP_{A1}), \tag{26}$$

where $\hat{t}$ is the direction of the translation (a unit vector) given by these methods. This extension is denoted with an * in the DOF part of the method name.

4.1. Synthetic Data

We generated two synthetic datasets of 10,000 random planar motions (R, t) each. Rotations are considered about the Y axis and thus the translations are on the $X-Z$ plane. For the first dataset (Y-ackermann), motions follow the Ackermann steering model, while the second dataset (Y-general) contains arbitrary translations on the plane. The 1pt-1*dof-2d2d method [32] is based on the Ackermann model and thus is not evaluated on the Y-general dataset. For the methods that return multiple solutions, we select the one that is closest to the ground truth orientation. We report the medians of the orientation error, computed as the angle of $R^\top R_{gt}$, the relative translation error, computed as $\|t-t_{gt}\|/\|t_{gt}\|$, where $_{gt}$ denotes the ground truth pose, and the plane of motion orientation error, as the angle between estimated and ground truth plane normal (i.e., rotation axis).

Pure rotation and translation. We verify our solver under pure rotation and translation. We take the Y-general dataset and randomly rotate each pose. Then, we set $t = 0$ for pure rotation and $R = I$ for pure translation, effectively generating two new datasets with 10K poses each. For pure rotation, maximum rotation, translation, and plane orientation errors are 3×10^{-6} degrees, 2×10^{-11} (absolute error), and 1×10^{-6} degrees, respectively. For pure translation, the maximum errors are 0 , 6×10^{-13} %, (plane is undefined).

Ground plane calibration error. We evaluate the effect of simulated ground plane calibration error (GPCE), attained by rotating the motions in the datasets in increments of 1 degree from -10 to 10 degrees. This preserves the orthogonality of the rotation axis and the translation, but the rotation axis is no longer aligned with the Y axis. The results for rotating the motions about the x axis are shown in Figure 4. For rotating about the z axis, the results are very similar and can be found on the supplementary material. Results highlight our method's differentiating property: by jointly estimating the motion parameters and the motion plane, our method obviates the need for ground plane calibration and is thus unaffected. In this regard, we clearly outperform the triplet of calibrated planar motion solvers [32, 6, 13] (i.e. assuming a motion plane parallel to the optical axis). As expected, 6-DOF methods are also unaffected by GPCE.

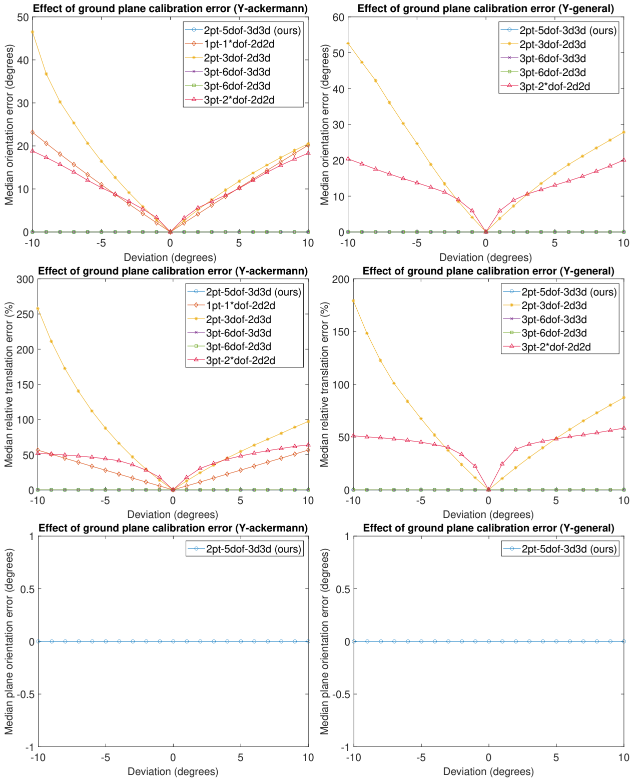

Figure 4. Effect of ground plane calibration error. Motions in the datasets were rotated about the x axis.

Non-planarity of motion. We evaluate performance for various degrees of non-planarity, attained by fixing the rotation and then rotating t about the axis orthogonal to both the rotation axis and t, in increments of 1 degree from -10 to 10 degrees. The results are shown in Figure 5. As expected, 6-DOF baselines are unaffected by non-planarity. All planar motion solvers are shown to have degraded performance as the magnitude of the deviation from planar motion increases. For deviations less than 6 degrees, our method consistently outperforms the 2pt-3dof-2d3d [6] and 3pt-2*dof-2d2d [13] calibrated planar methods. Alternative forms of motion planarity deviation that encode GPCE (i.e. translation is fixed and the rotation axis rotated about the axis orthogonal to both) yielded similar results and are in the supplementary material.

Measurement noise. First, we evaluate the effect of 2D pixel noise and depth noise independently. For 2D noise experiments, pixel measurements are disturbed by 2D Gaussian noise from 0 to 1 pixels in increments of 0.1 pixels and the corresponding depth value is unmodified. We consider a focal length of $f = 720$ pixels. The results are shown in Figure 6. We observe that our method is only outperformed by the 1pt-1*dof-2d2d method [32], which is only suitable under Ackermann motion, and the 3pt-6dof-3d3d non-

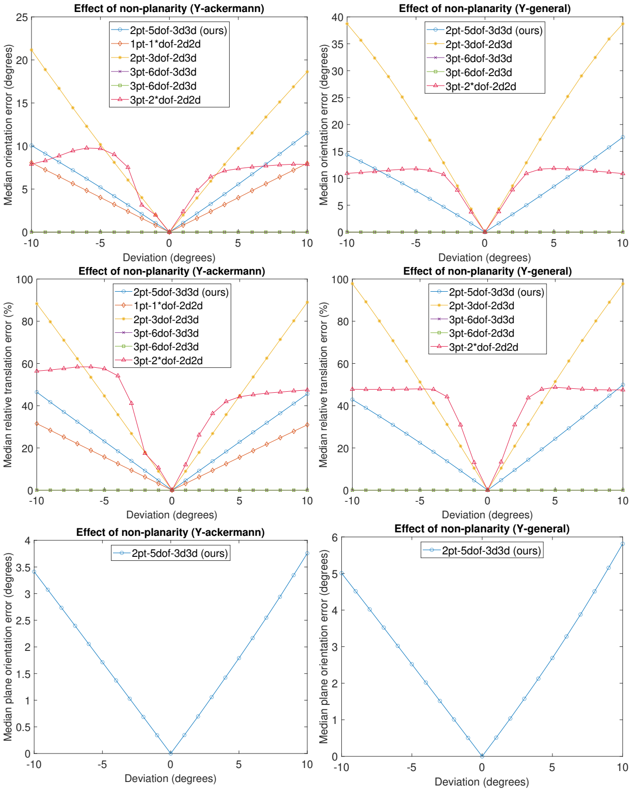

Figure 5. Effect of the non-planarity of motion when rotating t towards/away from the rotation axis.

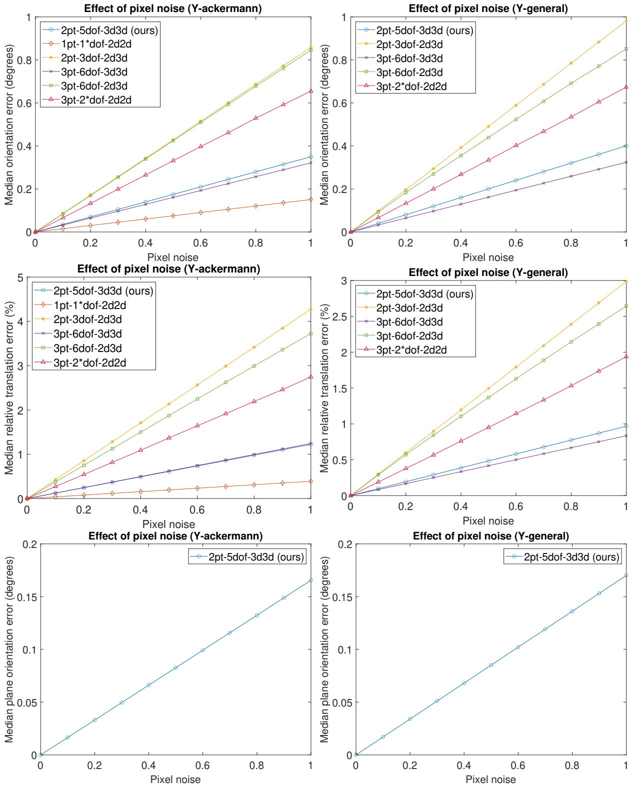

Figure 6. Effect of 2D pixel noise.

planar baseline [15]. The 2D-3D methods 2pt-3dof-2d3d [6] and 3pt-6dof-2d3d [28] are the most affected. For depth noise experiments, pixel measurements are fixed and depth is adjusted by $uS\%$ of its magnitude 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 7 where the plot's x axis corresponds to S. The 3D-3D methods 3pt-6dof-3d3d [15] and our 2pt-5dof-3d3d are the most affected because depth noise breaks the length invariance assumption between the camera frames. We note that the 2D-2D methods are unaffected by depth noise and the reason there is translation error for the 2D-2D methods is due to the extension we perform (Equation 26) to recover the scale of the translation from the (corrupted) 3D data. From these two experiments we observe that our method is more sensitive to depth errors than pixel errors. Finally, we evaluate the effect of 3D noise (Figure 8). The 3D coordinates are adjusted by $uS\%$ of their value, but now u is Gaussian with $\mu = 0$ and $\sigma = 1$. Our method shows comparable performance w.r.t. the 3pt-2*dof-2d2d [13], 3pt-6dof-2d3d [28], and 3pt-6dof-3d3d [15] methods for $S < 1.5$.

Execution time. We perform a relative evaluation of solver execution times on both synthetic datasets. All solvers were implemented completely in MATLAB (no pre-compiled mex). Solvers were run 30 times on each dataset for a total of $30 \times 10000 \times 2$ samples per solver. We report the median execution time for each solver in Table 1. Due to its simplicity, our method is faster than the others except for the 1pt-1*dof-2d2d Ackermann solver, which is about 5x faster. Compared to the remaining planar methods, our method achieves a speed up of ~5x w.r.t. 2pt-3dof-2d3d solver [6], and a speed up of ~9x w.r.t. the 3pt-2*dof-2d2d solver [13]. Compared to 6 DOF baselines, our solver is ~2x faster than 3pt-6dof-3d3d [15] and ~11x faster than 3pt-6dof-2d3d [28]. For methods returning multiple solutions, namely 3pt-2*dof-2d2d [13] and 3pt-6dof-2d3d [28], we do not include the time for selecting a single solution.

Table 1. Median execution time (μs) on the synthetic datasets. All methods implemented in MATLAB R2020b.

	Y-ackermann	Y-general
1pt-1*dof-2d2d [32]	**1.3**	**1.3**
2pt-5dof-3d3d (ours)	6.7	6.2
2pt-3dof-2d3d [6]	33.6	33.0
3pt-6dof-3d3d [15]	11.6	11.4
3pt-6dof-2d3d [28]	72.5	71.6
3pt-2*dof-2d2d [13]	59.6	59.2

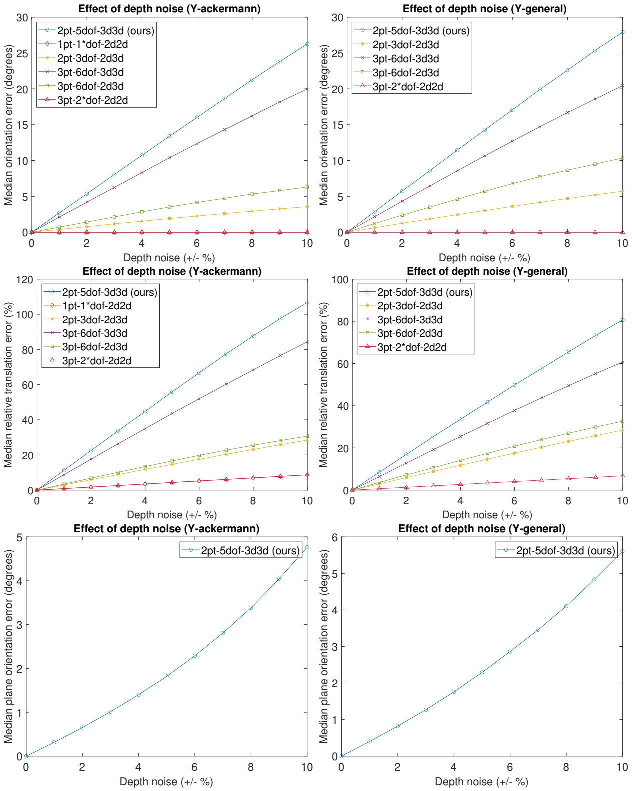

Figure 7. Effect of depth noise.

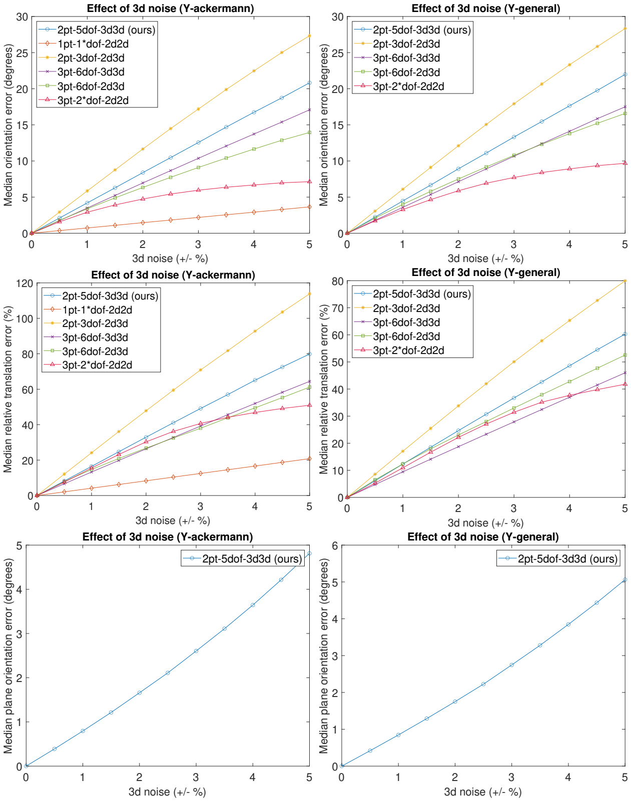

Figure 8. Effect of 3D noise.

4.2. Real Data

We evaluate our method's performance on two standard benchmarks: the KITTI odometry dataset [10] and the TUM RGB-D dataset [34]. For KITTI, we use the stereo pair corresponding to cameras 0 and 1 to extract 3D data. For each image, we extract SURF [1] features and match features across each stereo pair to obtain the corresponding 3D points via triangulation. Finally, we match the features across sequential pairs of images of camera 0 to obtain 3D-3D point correspondences. For TUM we also use SURF features and perform matching across sequential images to obtain 3D-3D point correspondences.

All solvers run inside a RANSAC framework with an adaptive number of iterations $N = \ln(1-p)/\ln(1-v^s)$, where we set $p = 0.9999$, v is the inlier ratio running estimate, and s the number of correspondences required per solver. Assuming $v = 0.5$ the number of iterations for the 1 point method [32] is 13, for the 2 point methods of ours and [6] the number is 32, and for the 3 point methods [15, 28, 13] the number is 69. The inlier threshold for 3D-3D point correspondences is a reprojection error of 3 pixels on both images. For solvers returning multiple solutions, all hypotheses are tested on all points.

We report the medians of the orientation error, computed as the angle of $R^\top R_{gt}$, and the translation error, computed as $\|t - t_{gt}\|$, where $_{gt}$ denotes the ground truth pose. We also report the median of the errors and of the success rate for all sequences in the ALL columns. The best result for each sequence is in bold and the best result of the planar methods is underlined. We consider a stride of 1 (i.e, pose is estimated for frames n and $n+1$). Results for other stride values can be found in the supplementary material.

KITTI dataset. For KITTI, we evaluate on the first 11 sequences, for which ground truth data is available. Due to space constraints, we only present results on the first 4 sequences in Table 2. The results on the remaining 7 sequences are in the supplementary material. We note that the median GPCE for all sequences is approximately 1.495 degrees and the median non-planarity error is 2.599 degrees. Our method outperforms the three calibrated planar solvers [32, 6, 13] in orientation estimation while the median translation error is within 2.2 cm of the best (planar) median error given by the 2pt-3dof-2d3d method [6]. We attribute the increase in translation error to our solver's sensitivity to 3D data discrepancies, as our 3D-3D solver is affected mainly by depth noise as shown in the synthetic experiments (Figure 7).The non-planar 3pt-6dof-2d3d method [28] yields the best results overall and the median errors of our method are within 0.086 degrees and 8.4 cm of it. Finally, for our solver the median plane orientation error for all sequences is 3.133 degrees (considering stride 5, since small strides approach pure translation).

Table 2. Median orientation error (in degrees), translation error (in meters), and success rate (in %) on the KITTI dataset [10] (stride of 1). Best in bold, best of planar methods underlined. The column ALL contains the median for all 11 sequences.

Sequence	00		01		02		03		ALL		
	deg	m	deg	m	deg	m	deg	m	deg	m	%
2pt-5dof-3d3d (ours)	0.186	0.105	0.143	1.274	0.181	0.116	0.177	0.125	0.168	0.116	**100.0**
3pt-6dof-3d3d [15]	0.255	0.129	0.359	2.025	0.270	0.147	0.242	0.152	0.245	0.147	**100.0**
2pt-3dof-2d3d [6]	0.237	0.109	0.112	0.378	0.198	0.094	0.182	0.094	0.175	0.094	**100.0**
3pt-6dof-2d3d [28]	**0.090**	**0.032**	**0.082**	**0.238**	**0.093**	**0.035**	**0.070**	**0.029**	**0.082**	**0.032**	**100.0**
1pt-1*dof-2d2d [32]	0.319	0.631	0.194	1.931	0.274	0.918	0.256	0.498	0.239	0.694	99.4
3pt-2*dof-2d2d [13]	0.247	0.130	0.134	0.654	0.206	0.114	0.200	0.110	0.188	0.114	**100.0**

Table 3. Median orientation error (in degrees), translation error (in meters), and success rate (in %) on the TUM dataset [34] (stride of 1). Best in bold, best of planar methods underlined. The column ALL contains the median for all 4 sequences.

Sequence	360		slam		slam2		slam3		ALL		
	deg	m	deg	m	deg	m	deg	m	deg	m	%
2pt-5dof-3d3d (ours)	0.755	0.059	0.560	0.027	0.500	0.025	0.355	0.018	0.530	0.026	98.9
3pt-6dof-3d3d [15]	1.035	0.087	0.724	0.038	0.621	0.037	0.460	0.026	0.673	0.037	**99.1**
2pt-3dof-2d3d [6]	0.562	0.048	0.526	0.030	0.438	0.023	0.300	0.015	0.482	0.027	95.5
3pt-6dof-2d3d [28]	0.659	0.033	0.517	0.017	0.462	0.015	0.300	0.012	0.490	0.016	98.9
1pt-1*dof-2d2d [32]	**0.408**	**0.015**	**0.379**	**0.013**	**0.337**	**0.009**	**0.229**	**0.007**	**0.358**	**0.011**	88.8
3pt-2*dof-2d2d [13]	0.440	0.029	0.404	0.017	0.356	0.012	0.246	0.009	0.380	0.015	96.1

TUM dataset. For TUM, we evaluate on the Robot SLAM sequences: 1) freiburg2 pioneer 360, 2) freiburg2 pioneer slam, 3) freiburg2 pioneer slam2, and 4) freiburg2 pioneer slam3. The results are in Table 3. Ackerman motion is well suited for this dataset, as the camera is right above the wheel axis (see Figure 1); in contrast to KITTI, where the distance between the camera and the back wheel axis is about 1 meter. We note that the median GPCE for all sequences is 1.509 degrees and the median non-planarity error is 0.595 degrees. Here, the 1pt-1*dof-2d2d [32] Ackermann solver yields the best results overall in orientation and translation estimation. Our method's median errors are within 0.172 degrees and 1.5 cm w.r.t. 1pt-1*dof-2d2d [32]. However, our success is significantly higher. It is also higher than the rates of the remaining calibrated planar methods [6, 13] and competitive to the rates of the 6-DOF methods [15, 28] (within 0.2%). For our solver, the median plane orientaton error for all sequences is 3.427 degrees (stride 10).

5. Conclusions

We introduced a geometric solver for relative camera pose from two 3D-3D point correspondences under planar motion. Our method is general in the sense that it does not require any knowledge about the motion plane, nor imposes an in-plane motion model. Moreover, given that the pose parameters are obtained from 3D data, the absolute scale of the translation is determined. Per limitations, our minimal solver lacks any mechanisms for evaluating and/or enforcing input data compliance to the planar motion assumption. Geometrically, we identify a degenerate case when observed 3D points are parallel to the rotation axis and a special case where the rotation cannot be uniquely determined from the planar motion constraint. However, such scenarios are rare in practice and are easily diagnosed by our formulation. Evaluation of our solver inside a RANSAC framework demonstrated effective outlier rejection and robust operation on real 3D data, with performance comparable to prior art but without the ground plane/motion limitations.

Acknowledgement

Work sponsored by the 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] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[2] Olaf Booij and Zoran Zivkovic. The planar two point algorithm. *Geojournal*, 2009.

[3] Sunglok Choi and Jong-Hwan Kim. Fast and reliable minimal relative pose estimation under planar motion. *Image and Vision Computing*, 69:103–112, 2018.

[4] Sunglok Choi, Jaeyeon Lee, Ji Hoon Joung, and Michael S. Ryoo. Numerical solutions to relative pose problem under

planar motion. In *The 7th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2010)*, 2010.

[5] Sunglok Choi, Jaehyun Park, and Wonpil Yu. Simplified epipolar geometry for real-time monocular visual odometry on roads. *International Journal of Control, Automation and Systems*, 13, 07 2015.

[6] Sung-In Choi and Soon-Yong Park. A new 2-point absolute pose estimation algorithm under planar motion. *Advanced Robotics*, 29:1–9, 05 2015.

[7] Chih Chung Chou and Chieh-Chih Wang. 2-point ransac for scene image matching under large viewpoint changes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3646–3651, 2015.

[8] Enrique Dunn, Brian Clipp, and Jan-Michael Frahm. A geometric solver for calibrated stereo egomotion. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 1187–1194, 2011.

[9] Jan-Michael Frahm, Jared Heinly, Enliang Zheng, Enrique Dunn, Pierre Fite-Georgel, and Marc Pollefeys. Geo-registered 3d models from crowdsourced image collections. *Geo-spatial Information Science*, 16(1):55–60, 2013.

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[11] T. Goedeme, T. Tuytelaars, L. Van Gool, G. Vanacker, and M. Nuttin. Feature based omnidirectional sparse visual path following. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1806–1811, 2005.

[12] Banglei Guan, Ji Zhao, Dániel Baráth, and Friedrich Fraundorfer. Minimal cases for computing the generalized relative pose using affine correspondences. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6048–6057, 2021.

[13] Levente Hajder and Daniel Barath. Least-squares optimal relative planar motion for vehicle-mounted cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8644–8650, 2020.

[14] Levente Hajder and Dániel Baráth. Relative planar motion for vehicle-mounted cameras from a single affine correspondence. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8651–8657, 2020.

[15] B. Horn. *Robot Vision*. MIT electrical engineering and computer science series. McGraw-Hill Book Company, 1986.

[16] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, Apr 1987.

[17] T. Ke and S. I. Roumeliotis. An efficient algebraic solution to the perspective-three-point problem. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4618–4626, July 2017.

[18] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, 2013.

[19] Bo Li, Evgeniy Martyushev, and Gim Hee Lee. 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*, page 215–231, Berlin, Heidelberg, 2020. Springer-Verlag.

[20] G. López-Nicolás, J.J. Guerrero, and C. Sagüés. Visual control of vehicles using two-view geometry. *Mechatronics*, 20(2):315–325, 2010.

[21] Zhixiang Min and Enrique Dunn. Voldor+ slam: For the times when feature-based or direct methods are not good enough. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13813–13819. IEEE, 2021.

[22] Zhixiang Min, Naji Khosravan, Zachary Bessinger, Manjunath Narayana, Sing Bing Kang, Enrique Dunn, and Ivaylo Boyadzhiev. Laser: Latent space rendering for 2d visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11122–11131, 2022.

[23] Zhixiang Min, Yiding Yang, and Enrique Dunn. Voldor: Visual odometry from log-logistic dense optical flow residuals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4898–4909, 2020.

[24] Zhixiang Min, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Enrique Dunn, and Manmohan Chandraker. Neurocs: Neural nocs supervision for monocular 3d object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21404–21414, 2023.

[25] Pedro Miraldo and Helder Araujo. Planar pose estimation for general cameras using known 3d lines. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4234–4240, 2014.

[26] Marcus Valtonen Örnhag and Anders Heyden. Relative pose estimation in binocular vision for a planar scene using inter-image homographies. In *ICPRAM*, pages 568–575, 2018.

[27] D. Ortín and J. M. M. Montiel. Indoor robot motion based on monocular images. *Robotica*, 19(3):331–342, may 2001.

[28] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (p3p) solver. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[29] Carolina Raposo and J. Barreto. 3d registration of curves and surfaces using local differential information. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9300–9308, 2018.

[30] Carolina Raposo and João P. Barreto. Using 2 point+normal sets for fast registration of point clouds with small overlap. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5652–5658, 2017.

[31] Olivier Saurer, Pascal Vasseur, Rémi Boutteau, Cédric Demonceaux, Marc Pollefeys, and Friedrich Fraundorfer. Homography based egomotion estimation with a common direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):327–341, 2017.

[32] Davide Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic

constraints. *International Journal of Computer Vision*, 95:74–85, 2011.

[33] Henrik Stewénius and Kalle Åström. Structure and motion problems for multiple rigidly moving cameras. In *ECCV*, volume 3023, pages 252–263, 05 2004.

[34] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[35] Xinfang Zhang, Yanyan Gao, Jian Chen, and Kaixiang Zhang. An efficient method to recover relative pose for vehicle-mounted cameras under planar motion. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(2):1138–1148, 2021.

[36] Ji Zhao, Laurent Kneip, Yijia He, and Jiayi Ma. Minimal case relative pose computation using ray-point-ray features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(5):1176–1190, 2020.