

# Robust Absolute and Relative Pose Estimation of a Central Camera System from 2D-3D Line Correspondences\*

Hichem Abdellali<sup>a</sup>, Robert Frohlich<sup>a</sup>  
<sup>a</sup>University of Szeged, Szeged, Hungary  
 hichem, frohlich@inf.u-szeged.hu

Zoltan Kato<sup>a,b</sup>  
<sup>b</sup>J. Selye University, Komarno, Slovakia  
 kato@inf.u-szeged.hu

## Abstract

*We propose a new algorithm for estimating the absolute and relative pose of a camera system composed of general central projection cameras such as perspective and omnidirectional cameras. First, we derive a minimal solver for the minimal case of 3 line pairs per camera, which is used within a RANSAC algorithm for outlier filtering. Second, we also formulate a direct least squares solver which finds an optimal solution in case of noisy (but inlier) 2D-3D line pairs. Both solver relies on Grobner basis, hence they provide an accurate solution within a few milliseconds in Matlab. The algorithm has been validated on a large synthetic dataset as well as real data. Experimental results confirm the stable and real-time performance under realistic outlier ratio and noise on the line parameters. Comparative tests show that our method compares favorably to the latest state of the art algorithms.*

## 1. Introduction

Absolute pose estimation of a camera consists in determining its position and orientation with respect to a reference 3D world coordinate frame, while relative pose estimation also aims to compute a rigid body transformation but with respect to another device (e.g. a reference camera), which is usually needed when a system of two or more sensors is considered. These are fundamental problems in a wide range of computer vision applications, such as visual

odometry, simultaneous localization and mapping (SLAM), image-based localization and navigation, augmented reality (AR) or structure-from-motion (SfM). The classical solutions focus only on a single perspective camera, and estimate pose using point correspondences. However, multi-camera systems containing a mixture of perspective and omnidirectional cameras are becoming more and more desirable by many modern applications, since they fit well the needs of special vision-based robotic and autonomous vehicle localization and navigation applications, where a higher field of view is often necessary for a robust interpretation of highly complex urban scenes.

Computer vision methods rely on the image content to establish correspondences needed to estimate the camera pose. The visual information can be of different complexity (e.g. points, lines, regions or even higher level semantic objects). Using  $n$  corresponding 2D-3D image points, called the *Perspective- $n$ -Point* (PnP) problem [7, 23, 24, 17, 15], is the most common absolute pose estimation approach, that can be solved with a minimum number of 3 correspondences. Various solutions exist for both large  $n$  as well as the  $n = 3$  minimal case (see [17] for a recent overview). However, point correspondences are less reliable in urban environment due to the repetitive structures of road scenes. Using line features instead of points is an attractive alternative in such scenarios, known as the *Perspective- $n$ -Line* (PnL) problem. A detailed overview of the available solutions was recently presented in [42]. Alternatively to point and line correspondences, there are also pose estimation methods relying on matching sets of 2D-3D regions [33, 32] or silhouettes [46]. More recently [29] proposed a more complex pipeline relying on point-to-plane mismatches, where image sets are registered to a 3D point-cloud through the use of structured scenes and polynomial sum-of-squares optimization framework. However, such approaches cannot be used in real-time driving and navigation applications due to their increased complexity.

The first globally optimal non-iterative solution for absolute pose of a single camera (AlgLS), proposed by [28], formulates the problem as a multi-variate polynomial system

\*This work was partially supported by the NKFI-6 fund through project K120366; "Integrated program for training new generation of scientists in the fields of computer science", EFOP-3.6.3-VEKOP-16-2017-0002; the Research & Development Operational Programme for the project "Modernization and Improvement of Technical Infrastructure for Research and Development of J. Selye University in the Fields of Nanotechnology and Intelligent Space", ITMS 26210120042, co-funded by the European Regional Development Fund; Research & Innovation Operational Programme for the Project: "Support of research and development activities of J. Selye University in the field of Digital Slovakia and creative industry", ITMS code: NFP313010T504, co-funded by the European Regional Development Fund.

with rotation parametrized by the Cayley-Gibbs-Rodriguez (CGR) representation. Zhang *et al.* proposed RPnL [45] which was further modified into the Accurate Subset-based PnL (ASPnL) method [42], which is one of the most accurate non-iterative methods. Another recent work from Wang *et al.* deals with the P3L [38] as well as with the PnL [39] problem. In [39], a fast and robust solution is proposed (called SRPnL) and its superior performance is confirmed by a comprehensive experimental comparison with many state of the art PnL solvers, like AlgLS [28], ASPnL [42]. Therefore in this work, we have validated our method through various experiments comparing it with SRPnL as it performed the best in [39] and AlgLS [28]. For multi-view camera systems, one notable work is the minimal NP3L solver of Lee [20], which deals with the 6 pose parameter estimation for a fully calibrated multi-view perspective camera system. In [12], the same problem is addressed with known vertical direction which leads to two fast and robust solvers, while in [1], absolute and relative pose is estimated with known vertical direction. While there are efficient solutions to line detection and matching in omnidirectional cameras [4], omnidirectional cameras are by far less researched in the context of line-based pose estimation. One solution with known vertical direction is proposed in [11]. While robust minimal solutions based on line correspondences for absolute pose [28, 45, 42, 20, 11, 39] or absolute and relative pose with known vertical direction [1] exists, none of these methods estimate full absolute and relative poses simultaneously in a multiview system including both omnidirectional and perspective cameras.

Since multi-view solutions are getting increasingly popular, especially in the autonomous driving [21], driver assistance with vehicle surrounding view [44, 43] and UAV mapping and monitoring domains, both the point and line based approaches to the multi-view pose estimation problem have been addressed (*e.g.* [27] uses the combination of the two). Solutions to the PnP or PnL problem cover only single-view perspective cameras, thus new methods were proposed that can efficiently deal with generalized PnP (gPnP) [40] or non-perspective PnP (NPnP) [5, 17, 21, 22] setups.

In this paper, we propose an universal solution for central camera setups, that provides a direct least squares solution to the absolute and relative pose problem. First, a minimal direct solver using Grobner bases is proposed which works with 3 line pairs, suitable for hypothesis testing in RANSAC [7]. Then a direct least squares solver is proposed which works for  $n \geq 3$  3D-2D line pairs. Both solvers run efficiently due to the low-order polynomial system of equations obtained via Cayley parametrization of the rotation matrix. The performance and robustness of the proposed method have been evaluated on large synthetic datasets as well as on real data with various camera systems, including different combinations of omnidirectional and perspective

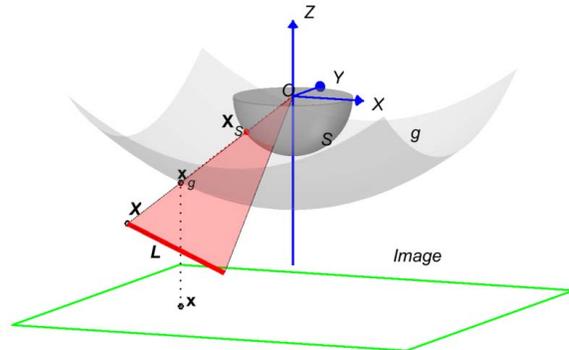


Figure 1. Projection plane of a line in the spherical camera model.

cameras.

## 2. Central Camera System

A unified model for central cameras was proposed by Geyer and Daniilidis [8], which represents central omnidirectional cameras as well as perspective cameras as a projection onto the surface of a unit sphere  $\mathcal{S}$ . The camera coordinate system is in the center of  $\mathcal{S}$ , and the  $Z$  axis is the optical axis of the camera which intersects the image plane in the *principal point*. This formalism has been adopted and models for the internal projection function have been proposed by Micusik [26, 25] and subsequently by Scaramuzza [31] who derived a general polynomial form  $g(\|\mathbf{x}\|) = a_0 + a_2\|\mathbf{x}\|^2 + a_3\|\mathbf{x}\|^3 + a_4\|\mathbf{x}\|^4$  which has 4 parameters representing the internal calibration parameters ( $a_0, a_2, a_3, a_4$ ) of the camera (only 4 parameters as  $a_1$  is always 0 [31]). Thus the nonlinear (but symmetric) distortion of central omnidirectional optics is represented by placing this rotationally symmetric  $g$  surface between the image plane and the unit sphere  $\mathcal{S}$  [31] (see Fig. 1). Knowing the internal calibration of the camera allows us to work directly with spherical image points  $\mathbf{x}_S \in \mathcal{S}$  using the bijective mapping of image points  $\mathbf{x} \mapsto \mathbf{x}_S$  composed of 1) lifting the image point  $\mathbf{x}$  onto the  $g$  surface by an orthographic projection

$$\mathbf{x}_g = \left[ \begin{array}{c} \mathbf{x} \\ a_0 + a_2\|\mathbf{x}\|^2 + a_3\|\mathbf{x}\|^3 + a_4\|\mathbf{x}\|^4 \end{array} \right] \quad (1)$$

and then 2) centrally projecting the lifted point  $\mathbf{x}_g$  onto the surface of the unit sphere  $\mathcal{S}$ :

$$\mathbf{x}_S = \frac{\mathbf{x}_g}{\|\mathbf{x}_g\|} \quad (2)$$

Similarly, the image points of a perspective camera can be represented on  $\mathcal{S}$  by the bijective mapping  $\mathbf{x} \mapsto \mathbf{x}_S$ :  $\mathbf{x}_K = \mathbf{K}^{-1}\mathbf{x}$  and  $\mathbf{x}_S = \mathbf{x}_K / \|\mathbf{x}_K\|$  (see Fig. 1). Thus the projection of a calibrated central camera is fully described

by means of unit vectors  $\mathbf{x}_S$  in the half space of  $\mathbb{R}^3$ . A 3D world point  $\mathbf{X}$  is projected into  $\mathbf{x}_S \in \mathcal{S}$  by a simple central projection taking into account the pose:

$$\mathbf{x}_S = \frac{\mathbf{R}\mathbf{X} + \mathbf{t}}{\|\mathbf{R}\mathbf{X} + \mathbf{t}\|} \quad (3)$$

## 2.1. Projection of Lines and Camera Pose

Let us now see how a camera system composed of  $N$  central cameras [17, 5, 30, 21] projects 3D lines. Note that a central camera may or may not be perspective! Even when a camera has a single effective viewpoint, its projection model may include non-linear distortions, like in the case of central omnidirectional cameras [3, 8, 13, 31, 32]. Herein, we will consider an arbitrary mixture of perspective and non-perspective central cameras and derive unified equations for such a camera system. In this paper, 3D lines will be represented as  $L = (\mathbf{V}, \mathbf{X})$ , where  $\mathbf{V}$  is the unit direction vector of the line and  $\mathbf{X}$  is a point on the line [34, 11, 1]. A 3D line  $L$  is centrally projected by a projection plane  $\pi_L = (\mathbf{n}, d)$  onto the surface  $\mathcal{S}$ . Since the camera projection center is also on  $\pi_L$ ,  $d$  becomes zero and thus  $\pi_L$  is uniquely determined by its unit normal  $\mathbf{n}$ . The image of  $L$  is the intersection of the ray surface  $\mathcal{S}$  and  $\pi_L$ , which is a *great circle*, while a particular line segment becomes a *great circle segment* on the unit sphere  $\mathcal{S}$  with endpoints  $\mathbf{a}$  and  $\mathbf{b}$  (both are on  $\mathcal{S}$ , hence they are unit length!). The unit normal  $\mathbf{n}$  to the projection plane  $\pi_L$  in the camera coordinate frame  $\mathcal{C}$  is then given by  $\mathbf{n} = \mathbf{a} \times \mathbf{b}$ . Since  $L$  lies also on  $\pi_L$ , its direction vector  $\mathbf{V}$  is perpendicular to  $\mathbf{n}$ . Hence we get the following equation which involves only the absolute pose  $(\mathbf{R}, \mathbf{t})$  [11, 1]

$$\mathbf{n}^\top \mathbf{R}\mathbf{V} = \mathbf{n}^\top \mathbf{V}^C = 0, \quad (4)$$

where  $\mathbf{R}$  is the rotation matrix from the world  $\mathcal{W}$  to the camera  $\mathcal{C}$  frame and  $\mathbf{V}^C$  denotes the unit direction vector of  $L$  in the camera coordinate frame  $\mathcal{C}$ . Furthermore, the vector from the camera center  $\mathbf{C}$  to the point  $\mathbf{X}$  on line  $L$  is also lying on  $\pi_L$ , thus it is also perpendicular to  $\mathbf{n}$ :

$$\mathbf{n}^\top (\mathbf{R}\mathbf{X} + \mathbf{t}) = \mathbf{n}^\top \mathbf{X}^C = 0, \quad (5)$$

where  $\mathbf{t}$  is the translation from the world  $\mathcal{W}$  to the reference camera  $\mathcal{C}$  frame and  $\mathbf{X}^C$  denotes the point  $\mathbf{X}$  on  $L$  in the camera coordinate frame  $\mathcal{C}$ .

When we have  $N$  central cameras, a 3D line  $L$  has up to  $N$  images, one in each camera. These cameras may be assembled into an ad-hoc multi-camera system or they might originate from a single camera moving along a trajectory [30, 5, 21, 2] – in either case, they form a camera system with unknown relative poses  $(\mathbf{R}_i, \mathbf{t}_i) : \mathcal{C} \rightarrow \mathcal{C}_i$  with respect to the reference camera coordinate frame  $\mathcal{C}$ . The projection of  $L$  in these relative cameras yield similar equations as (4) and (5) but the relative pose  $(\mathbf{R}_i, \mathbf{t}_i)$  will also

be involved:

$$\mathbf{n}_i^\top \mathbf{R}_i \mathbf{V}^C = \mathbf{n}_i^\top \mathbf{R}_i \mathbf{R}\mathbf{V} = 0 \quad (6)$$

$$\mathbf{n}_i^\top (\mathbf{R}_i \mathbf{X}^C + \mathbf{t}_i) = \mathbf{n}_i^\top (\mathbf{R}_i (\mathbf{R}\mathbf{X} + \mathbf{t}) + \mathbf{t}_i) = 0 \quad (7)$$

The pose (either absolute or relative) has 6 degrees of freedom. Thus to solve for either the absolute pose using (4) and (5) or the relative pose using (6) and (7), we need a minimum of 3 line correspondences. The solution is obtained in two steps: first the rotation is solved using (4) (or (6)), which in general involves solving a system of 8-th order polynomials [6]. Then translation is obtained from (5) (or (7)) by backsubstituting the rotation, which yields a linear system of equations in terms of the translation [42, 20, 45, 38]. Clearly, the main challenge is the solution for rotation due to the nonlinearity of the equations as well as the additional constraints to ensure a valid rotation (*i.e.* orthonormal) matrix. Although for special line configurations (*e.g.* orthogonal, parallel or intersecting lines) [42] or with additional knowledge of *e.g.* the vertical direction [11, 1], a lower order polynomial may be achieved, most of the P3L polynomials proposed in the literature will not be lower than 8 for general line configurations [6, 42, 20].

## 2.2. Cayley Parametrization of 3D Rotations

Let us have a closer look at the parametrization of the rotation matrix  $\mathbf{R}$ . It is well known, that the rotation group  $SO(3)$  has 3 degrees of freedom. The most popular parametrization is *Euler angles*, which defines a rotation in terms of three consecutive elemental rotations around the orthogonal axes  $X$ ,  $Y$ , and  $Z$  of a Cartesian coordinate system. Since it involves trigonometric functions, this representation would yield trigonometric equations. To get rid of these trigonometric functions, one common approach is to letting these trigonometric functions to be two separate unknowns [24, 42, 45, 11], which –together with the trigonometric constraints– leads to polynomial equations. Alternatively, one can solve directly for the 9 elements of  $\mathbf{R}$  in (4) –as a linear system– and then enforce orthonormality on the solution yielding again to (different) polynomial equations [39, 42].

Of course, these parametrizations directly influence the type of equations one can derive from (4) for computing  $\mathbf{R}$ . Herein, we will use the Cayley transform to obtain a parametrization of the rotation matrix  $\mathbf{R}$  in terms of 3 parameters  $\mathbf{b} = [b_1, b_2, b_3]^\top$ . Following [41, 9], The Cayley transform of a rotation matrix is a skew-symmetric matrix and vice versa. Therefore the correspondence  $\mathbf{R} \leftrightarrow [\mathbf{b}]_\times$  is a one-to-one map between skew-symmetric matrices (represented as 3-vectors) and 3D rotations, excluding rotation

angles  $\pm 180^\circ$ . Thus we have

$$(1 + \mathbf{b}^\top \mathbf{b})\mathbf{R} = (1 - \mathbf{b}^\top \mathbf{b})\mathbf{I} + 2[\mathbf{b}]_\times + 2\mathbf{b}\mathbf{b}^\top = \begin{bmatrix} 1 + b_1^2 - b_2^2 - b_3^2 & 2b_1b_2 - 2b_3 & 2b_1b_3 + 2b_2 \\ 2b_1b_2 + 2b_3 & 1 - b_1^2 + b_2^2 - b_3^2 & 2b_2b_3 - 2b_1 \\ 2b_1b_3 - 2b_2 & 2b_2b_3 + 2b_1 & 1 - b_1^2 - b_2^2 + b_3^2 \end{bmatrix} \quad (8)$$

Note that in the equations (4) and (6), we only use the above matrix, but to get the proper rotation matrix  $\mathbf{R}$  from  $\mathbf{b}$ , the scale factor  $(1 + \mathbf{b}^\top \mathbf{b})$  has to be used too!

### 3. Minimal Solver

Given a set of 2D-3D putative line correspondences, first an inlier set has to be determined in order to obtain a robust pose estimate. This can be done via RANSAC [7] or the M-estimator sample consensus (MSAC) algorithm [35], which relies on a minimal solver and a backprojection error metric. In our case, the minimal set consists of 3 line-pairs, providing 3 equations for the rotation only as in (4) and 3 equations for the translation as in (5). Using the Cayley parametrization of the rotation matrix  $\mathbf{R}$ , we get the following second order polynomial equation from (4):

$$\mathbf{c}^\top \mathbf{x} = \begin{bmatrix} n_1v_1 + n_2v_2 + n_3v_3 \\ 2n_1v_2 + 2n_2v_1 \\ 2n_1v_3 + 2n_3v_1 \\ 2n_2v_3 + 2n_3v_2 \\ n_1v_1 - n_2v_2 - n_3v_3 \\ -n_1v_1 + n_2v_2 - n_3v_3 \\ -n_1v_1 - n_2v_2 + n_3v_3 \\ -2n_2v_3 + 2n_3v_2 \\ 2n_1v_3 - 2n_3v_1 \\ -2n_1v_2 + 2n_2v_1 \end{bmatrix}^\top \begin{bmatrix} 1 \\ b_1b_2 \\ b_1b_3 \\ b_2b_3 \\ b_1^2 \\ b_2^2 \\ b_3^2 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \mathbf{0} \quad (9)$$

where  $\mathbf{n} = [n_1, n_2, n_3]^\top$  and  $\mathbf{V} = [v_1, v_2, v_3]^\top$  are the projection plane unit normal and the 3D line unit direction vector, respectively. Given 3 such line-pairs, we obtain a system of 3 equations of the form (9) in the 3 unknown rotation parameters  $\mathbf{b} = [b_1, b_2, b_3]^\top$ , which can be easily solved by a solver using Grobner basis [18, 16, 19]. In our experiments, we used the automatic generator of Kukulova *et al.* [18] for a fair comparison in Matlab with competing methods, but we remark that we also successfully used Kneip's generator [16] which produces a solver in C++, that is an order of magnitude faster! The translation  $\mathbf{t}$  is then obtained by backsubstituting  $\mathbf{R}$  into (5) yielding a system of linear equations, which can be solved by SVD decomposition. Although (9) might have several solutions, the solver will only return the real ones and then one has to select the geometrically valid  $(\mathbf{R}, \mathbf{t})$  based on the visibility of the lines and the backprojection error (see Section 3.1). Note that the relative camera poses  $(\mathbf{R}_i, \mathbf{t}_i)$  are also obtained in a

similar way once the absolute pose  $(\mathbf{R}, \mathbf{t})$  is computed and backsubstituted into (6) and (7).

### 3.1. Line Backprojection Error on the Unit Sphere

RANSAC will iteratively sample a minimal line-set, solve it via the minimal solver outlined above, and then classify the line-pairs into inliers and outliers based on the backprojection error. Therefore the second component for our robust pose estimation is the backprojection error. While solutions exist for perspective cameras [20, 34], these metrics are not usable in our case as we are working on the unit sphere  $\mathcal{S}$ . Therefore, given an observed image line segment with its endpoint spherical coordinates  $(\mathbf{a}, \mathbf{b})$  and the corresponding 3D line backprojected to the unit sphere as  $(\mathbf{A}, \mathbf{B})$ , let us define the backprojection error of the 3D line w.r.t. its observed image line directly on  $\mathcal{S}$ . First of all, lines become *great circles* and a particular line segment becomes a *great circle segment* on the unit sphere. We will derive an error function to characterize the line projection error as the distance between  $(\mathbf{a}, \mathbf{b})$  and  $(\mathbf{A}, \mathbf{B})$ . Points on the observed spherical line  $(\mathbf{a}, \mathbf{b})$  can be parametrized by a scalar  $\phi = 0, \dots, \lambda$  with  $\lambda$  being the geodesic length (or *great-circle length* or *orthodromic length* of the observed segment:

$$\lambda = \arctan \left( \frac{\|\mathbf{a} \times \mathbf{b}\|}{|\mathbf{a} \cdot \mathbf{b}|} \right). \quad (10)$$

A simple metric is to compute the shortest *orthodromic distance* of the endpoints of the observed segment to the backprojected line  $(\mathbf{A}, \mathbf{B})$ . Let  $\delta(\mathbf{p})$  be the shortest *orthodromic distance* of a point  $\mathbf{p}$  on the observed segment to the backprojected line  $(\mathbf{A}, \mathbf{B})$ :

$$\begin{aligned} \delta(\mathbf{p}) &= \arctan \left( \frac{|\mathbf{n} \cdot \mathbf{p}|}{\|\mathbf{n} \times \mathbf{p}\|} \right) \\ &= \arctan \left( \frac{\|\mathbf{n}\| \|\mathbf{p}\| |\sin \theta|}{\|\mathbf{n}\| \|\mathbf{p}\| |\cos \theta|} \right) = \theta \end{aligned} \quad (11)$$

where  $\mathbf{n}$  is the unit normal vector of the projection plane of  $(\mathbf{A}, \mathbf{B})$ , *i.e.*

$$\mathbf{n} = \frac{\mathbf{A} \times \mathbf{B}}{\|\mathbf{A} \times \mathbf{B}\|}, \quad (12)$$

$\theta$  is the angle (in radian) between  $\mathbf{p}$  and the plane with normal  $\mathbf{n}$ , *i.e.* the plane passing through  $\mathbf{A}$ ,  $\mathbf{B}$ , and the center of the sphere. Thus the shortest distance of  $\mathbf{a}$  and  $\mathbf{b}$  to the backprojected line are given by  $\delta(\mathbf{a})$  and  $\delta(\mathbf{b})$ . Since the error represented by these distances is inversely proportional to the length of the line segment (same  $\delta$  distance on a longer line segment means a smaller backprojection error), we used sum of squared distances weighted with the inverse of the length  $\lambda$  of the line segment:

$$\frac{1}{\lambda} (\delta^2(\mathbf{a}) + \delta^2(\mathbf{b})). \quad (13)$$

### 3.2. Normalization

The 2D image data is normalized by definition as we work on the unit sphere. However, the 3D lines are given in an arbitrary world coordinate system  $\mathcal{W}$ , which needs to be normalized for numerical stability [10]. Herein, we thus transform our 3D line segments into a unit cube around the origin of  $\mathcal{W}$ : First a uniform scaling factor  $s$  is calculated using the maximum domain of the coordinates along all three axes, practically the height  $h$ , width  $w$  and depth  $d$  of the data, then choosing the maximum of these as the uniform scaling measure  $s = \frac{1}{\max(|h|, |w|, |d|)}$ . We also calculate the centroid  $[u, v, z]^T$  of the 3D scene points. Then the normalization matrix  $\mathbf{N}$  is composed of the translation  $-[u, v, z]^T$  followed by a uniform scaling by  $s$ . We used uniform scaling to avoid changing the direction vector of the 3D line. This way, the rotational part of the pose is not affected by this normalization. Normalization is then applied to the 3D line points  $\mathbf{N}\mathbf{X}$  used in (5) and (7). The solution is then obtained in this normalized space, hence the result  $(\tilde{\mathbf{R}}, \tilde{\mathbf{t}})$  need to be denormalized. Since the equations used to solve for the rotation are unaffected by this normalization (thanks to uniform scaling!),  $\tilde{\mathbf{R}}$  is the final rotation, while the translation  $\tilde{\mathbf{t}}$  needs to be corrected by applying  $\begin{bmatrix} \mathbf{I} & \tilde{\mathbf{t}} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{N}$ .

### 4. Direct Least Squares Solver

Let us now focus on the general case, when we have  $n > 3$  inlier but noisy 2D-3D line pairs. We also start from (4) and (5). Each line pair generates one such pair of equations, yielding a system of  $n > 3$  equations, which is solved in the least squares sense. For this purpose, let's take the sum of squares of the nonlinear system constructed from (9):

$$E(\mathbf{b}) = \sum_{i=1}^n (\mathbf{c}_i^T \mathbf{x})^2 \quad (14)$$

and then find  $\arg \min_{\mathbf{b}} E(\mathbf{b})$ . The first order optimality condition is

$$\nabla E(\mathbf{b}) = \begin{bmatrix} \frac{\partial E(\mathbf{b})}{\partial b_1} \\ \frac{\partial E(\mathbf{b})}{\partial b_2} \\ \frac{\partial E(\mathbf{b})}{\partial b_3} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n \mathbf{d}_{\mathbf{b}_1 i}^T \mathbf{x}_{\mathbf{b}_1} \\ \sum_{i=1}^n \mathbf{d}_{\mathbf{b}_2 i}^T \mathbf{x}_{\mathbf{b}_2} \\ \sum_{i=1}^n \mathbf{d}_{\mathbf{b}_3 i}^T \mathbf{x}_{\mathbf{b}_3} \end{bmatrix} = \mathbf{0} \quad (15)$$

where for each line pair  $\mathbf{d}_{\mathbf{b}_1}$ ,  $\mathbf{d}_{\mathbf{b}_2}$ , and  $\mathbf{d}_{\mathbf{b}_3}$  can be expressed in terms of the coefficients  $\mathbf{c}$  of each line pair. Thus the solution of the system of 3 polynomial equations (each of them is third order) in (15) provides the rotation parameters  $\mathbf{b}$ . We successfully used the solver generator of [18] to generate a Matlab solver for the above polynomial system. The translation  $\mathbf{t}$  is then obtained by backsubstituting  $\mathbf{R}$  into (5) yielding a system of linear equations, which can be solved

by SVD decomposition. Multiple solutions are eliminated in the same way as for the minimal solver. Relative camera poses  $(\mathbf{R}_i, \mathbf{t}_i)$  are also obtained in a similar way once the absolute pose  $(\mathbf{R}, \mathbf{t})$  is computed and backsubstituted into (6) and (7).

---

**Algorithm 1** Summary of the proposed robust pose estimation algorithm for  $N$  central cameras

---

**Input:** 3D-2D putative line matches from  $N$  cameras

**Output:** The absolute pose  $(\mathbf{R}, \mathbf{t}) : \mathcal{W} \rightarrow \mathcal{C}$  and the relative poses  $(\mathbf{R}_i, \mathbf{t}_i) : \mathcal{C} \rightarrow \mathcal{C}_i$

- 1: Calculate  $\mathbf{N}$  as in Section 3.2 and normalize the 3D line endpoints  $\mathbf{X}$ .
  - 2: Calculate the normal  $\mathbf{n}$  of the projection plane for each 2D line and the unit direction vector  $\mathbf{V}$  for each 3D line  $\mathbf{L}$  as described in Section 2.1.
  - 3: Filter outliers using the Cayley minimal solver proposed in Section 3 with MSAC.
  - 4: Using the obtained inlier set of 2D-3D line pairs estimate the absolute  $(\tilde{\mathbf{R}}, \tilde{\mathbf{t}})$  and relative  $(\tilde{\mathbf{R}}_i, \tilde{\mathbf{t}}_i)$  poses with the Cayley-LS solver presented in Section 4.
  - 5: Return the denormalized  $(\mathbf{R}, \mathbf{t})$  and  $(\mathbf{R}_i, \mathbf{t}_i)$  poses.
- 

### 5. Experimental Results

Quantitative evaluation was performed on synthetically generated datasets. Since for both the perspective and omnidirectional cameras we used the calibration parameters of real cameras, with available physical parameters such as the sensor size, we calculated an estimated pixel-to-meter ratio, thus being able to represent our 3D scene in an equivalent metric space. Multiple sets of 1000 samples were generated containing 3D-2D line pairs. The 3D scene was created with a typical road scene in mind, where only a few planar surfaces are usually visible in a camera, thus we created 3 planes randomly placed (with a rotation of  $\pm 30^\circ$  around all 3 axes,  $\pm[1 - 2]$  m horizontal and vertical translation, and  $\pm[0.5 - 1.5]$  m in depth) in the 3D space, each containing 20 random line segments, with a minimum length of 0.5 m.

For the 2D side we generated images of the scene by projecting the lines with perspective and omnidirectional cameras as well, using the parameters of a standard commercial camera with APS-C size sensor, 2378x1580 pixel resolution and 16 mm normal lens, respectively an 8 mm fisheye lens. Each camera was placed in the scene with a random rotation of  $\pm 50^\circ$  around all 3 axes, and random translation of  $\pm 1$  m in the horizontal and vertical direction, while in the optical axis' direction the perspective camera was placed at  $[4 - 6]$  m from the scene, the omnidirectional camera at  $[2 - 3]$  m. Each of the 1000 test cases contains 2+1 cameras, both in the omni-perspective and perspective-omni configuration to cover all possible variations of reference and relative cameras.

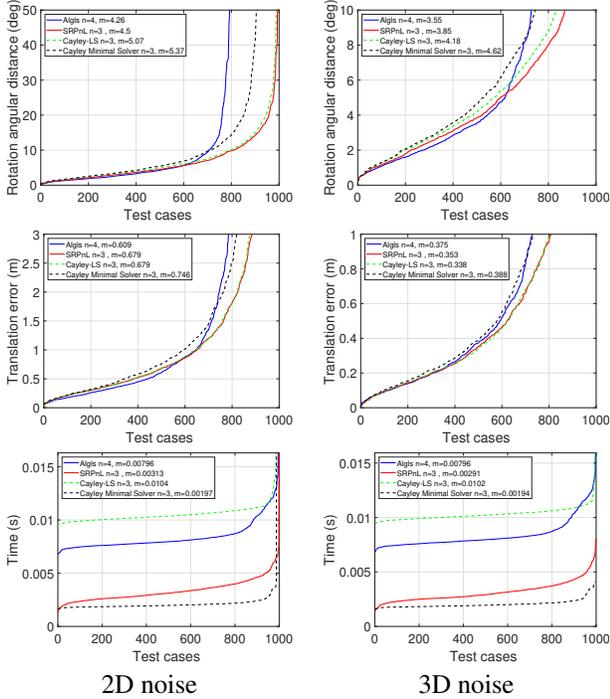


Figure 2. Comparison to State-of-the-Art methods with 7% 2D (left) and 7% 3D noise (right) using the minimum number of line matches with each method.

To evaluate the robustness of the proposed algorithm for noisy line detections, we simulated noise by corrupting one endpoint of the line (similarly in 2D and 3D), essentially adding a random number to each coordinate of the point up to the specified percentage of the actual coordinate value. The unit direction vector was also modified in the same manner. We show results for 7% and 15% 2D and 3D noise levels, with the only exception that on the omnidirectional images the 2D noise was limited to 10% instead of 15% because of the high nonlinear distortion of the camera that accentuates these errors. These error levels translate to a shift on the image with an average of 55 – 51px (with 7% omni and perspective noise), 79px (with 10% omni noise) and 110px (with 15% perspective noise) respectively.

### 5.1. Comparison with State-of-the-Art

The proposed Cayley minimal solver and Cayley least squares solver (Cayley-LS), were compared to two State-of-the-Art methods: AlgLS, one of the most accurate non-iterative methods, which estimates the camera’s pose by directly solving the corresponding least-squares problem algebraically, and SRPnL, a novel closed-form solution to the PnL pose problem that solves univariate polynomials and includes a Gauss Newton refinement. Comparisons were performed in two different setups, first using the minimum number of line matches that each algorithm requires, then using all 60 line pairs of the scene, only on a single per-

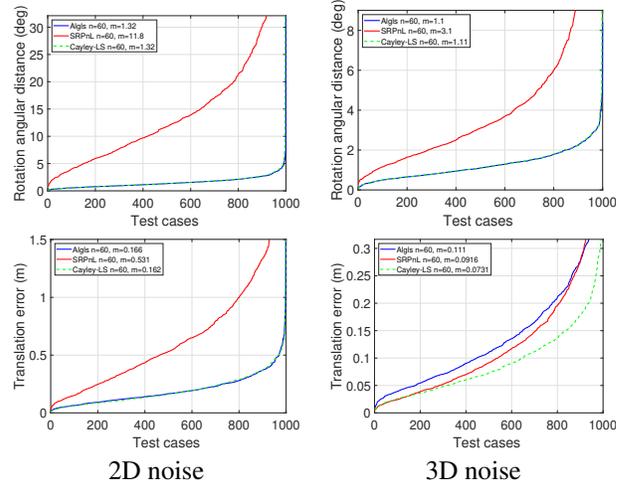


Figure 3. Comparison to State-of-the-Art methods with 15% 2D noise on the left and 15% 3D noise on the right, using  $n = 60$  line pairs.

spective camera, since the formulation of the other methods doesn’t support omnidirectional cameras, neither a multi-view setup was presented with them.

The minimum number of lines required is  $n=3$ , except for AlgLS, that uses  $n=4$ . Based on Fig. 2 we can conclude that all the methods perform very similar in terms of median errors of the pose parameters, only AlgLS produces slightly lower median errors due to the higher number of line-pairs it is using ( $n=4$ ), but it is also the least robust of all tested methods, producing higher than  $20^\circ$  rotation error in 25% of the cases, compared to 15% with Cayley Minimal Solver, and 5% with SRPnL and Cayley-LS. None of the methods are handling well this level of noise, median angular distance is above  $3.5^\circ$  and  $4.2^\circ$  and translation error above 35 cm and 60 cm for 3D and 2D noise respectively. In terms of runtime, the proposed Cayley minimal solver is the fastest with 2 ms, followed by SRPnL with 3 ms, then AlgLS with 8 ms and Cayley LS 10 ms.

We also performed comparisons in case of  $n = 60$  line pairs, where we also obviously excluded from the comparisons shown in Fig. 3 the Cayley Minimal Solver, since we are using  $n = 60$  lines. The error plots in Fig. 3 show that AlgLS and Cayley-LS have the best results with lowest median rotation and translation errors, robust for up to 15% noise with median angular distance below  $1.5^\circ$ , but AlgLS favors the noise in 2D domain. SRPnL shows lack of precision, producing much higher pose errors, already with 7% noise. Cayley-LS has the same characteristic as AlgLS in every test setup, the only difference noticeable is in case of 15% 3D noise where Cayley-LS is more robust in terms of translation error (see Fig. 3), while practically having the same execution time of 9 ms as AlgLS. With 60 lines the runtime of the algorithms keep the same characteristic and very similar median value, the number of lines not affecting

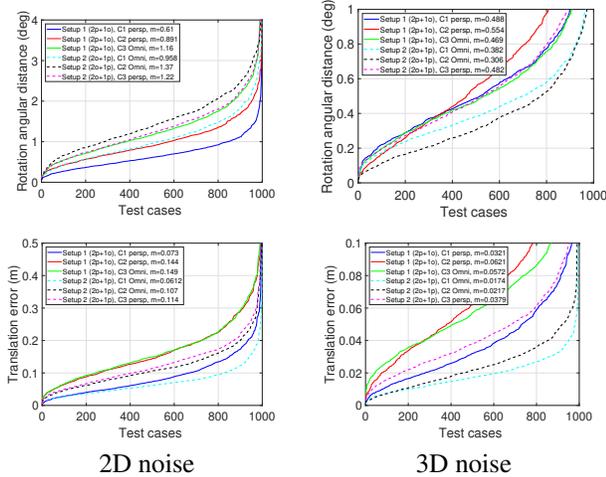


Figure 4. Comparison of Cayley-LS results between different camera compositions in terms of median pose errors with 7% 2D noise on the left and 7% 3D noise on the right.

significantly the runtime.

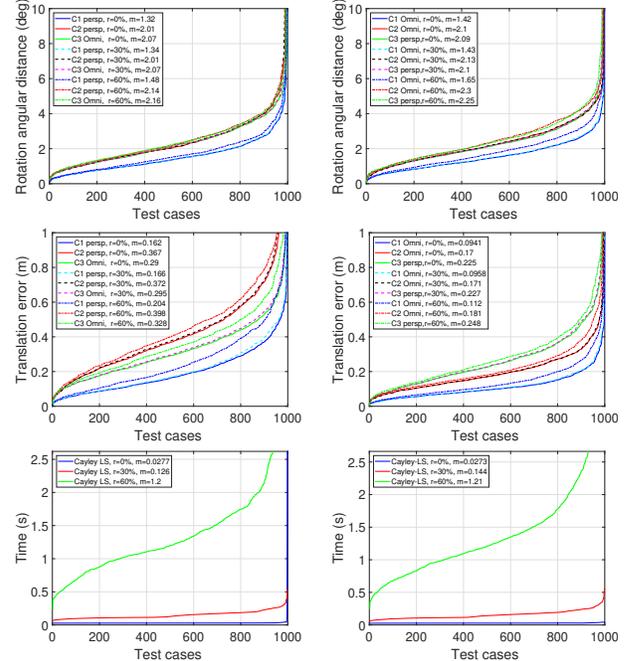
We remark, that for the purpose of a fair evaluations we used the Matlab solver generated by [18], while also a C++ solver generated by the automatic tool from the Polyjam software [16] is available, which is much faster.

## 5.2. Multi-view Setup Composition

Since we are dealing with perspective and omnidirectional cameras in the same framework, we have to test the sensibility of the proposed algorithms to different compositions of the camera setup. Due to the way our equations are formalized, we always have a selected reference camera that can propagate errors to the other cameras due to the absolute and relative pose formulation. In Fig. 4 we can see the two relevant configurations with 7% noise, *setup 1* referring to 2 perspective and one omni ( $2p+1o$ ) cameras while *setup 2* referring to 2 omni and 1 perspective ( $2o+1p$ ) cameras. These setups contain all 4 possible combinations of reference and relative camera. Based on Fig. 4 we can conclude, that there is no clear advantage of using one type of camera or the other as a reference, since *setup 1* has slightly better results with 2D noise, while *setup 2* is better with 3D noise.

## 5.3. Robustness to Outliers

Since the proposed Cayley minimal solver proved to be the fastest of the tested methods and robust to noise, it is well suited for outlier detection in a RANSAC algorithm. In our experiments we used the built in M-estimator sample consensus (MSAC) algorithm function of Matlab [35] together with the backprojection error presented in Section 3.1. The synthetic dataset previously defined was extended by adding a specific number of outlier 2D-3D line-pairs with randomly generated coordinates, to obtain the outlier ratio of: 30% and 60% (26, 90 outliers respectively).



*setup 1* with ( $2p+1o$ )

*setup 2* with ( $2o+1p$ )

Figure 5. Cayley-LS pose estimation results on the inlier line-pairs provided by RANSAC with 15% noise and  $r = 30\%$ ,  $r = 60\%$  outlier ratio, compared to the baseline results without RANSAC on the inlier set  $r = 0\%$ .

The threshold for RANSAC was experimentally determined as the average between the maximum of the inliers' and minimum of the outliers' backprojection error calculated with the reference pose. In our tests RANSAC with the Cayley minimal solver was able to robustly filter out all outliers, since there was a clear separation between the inliers and outliers, but we found that a smaller inlier set can only be obtained if the outlier lines are taken from the same planes as the inliers, thus they are not different enough from the correct lines. Pose estimation errors of the Cayley-LS solver on the inlier sets, using the two camera configurations presented previously with 15% noise and 30%, 60% outlier ratio, are shown in Fig. 5. We can see that the algorithm is robust up to 15% noise with 60% outliers where the angular distance or the translation doesn't change too much compared to the Cayley-LS solver run only on inliers. The runtime plots in the last row of Fig. 5 show that, while a higher than 50% outlier ratio can be filtered out robustly, it drastically increases the execution time of the algorithm. If a reduced number of outliers can be assumed ( $< 30\%$  outlier ratio) a 5 folds increase in runtime is to be expected, that could still fit in many applications' requirements.

## 5.4. Real Data

To evaluate the proposed algorithm on real data, we have used a set of 17 2D perspective and omnidirectional im-

ages captured in an outdoor urban environment, where the dense 3D point cloud of the scene was captured with a Riegl VZ400 Lidar scanner with an angular resolution of  $0.05^\circ$ . The perspective images were captured by a flying drone in 4K resolution, while the omnidirectional images were taken with a Canon DSLR camera with a 8mm fisheye lens. The ground truth pose of each camera images was estimated with UPnP [17] using highly reflective markers placed on the building surface, that were automatically scanned by the scanner, and detected and matched with 2D manually. Relying on these markers we can compute a metric forward projection error to evaluate the precision of the estimated camera pose. For the reference poses the maximum forward projection error was 10 cm, and the median was 3 cm. To provide the necessary input to our algorithm, we detected 2D lines on the perspective images using the OpenCV LSD detector [37], while on the omnidirectional images we used the automatic line extraction toolbox of Bermudez [4]. The corresponding 3D lines were produced by relying on the images captured with the camera attached to the scanner, that has a very precise pose calibrated in laboratory environment. We used these to project 2D lines detected on the reference images into the 3D pointcloud, then manually matching the 2D lines from our evaluation set of 17 images with the lines on the reference images, we directly obtained the 2D-3D matches. This could also be done by calculating line segment descriptors like [36] and using them to automatically match lines.

We evaluated the Cayley-LS solver on the 17 images of the real dataset by choosing randomly a reference camera. The results show, that despite the fact that we used only a relatively small number of lines (an average of 15 lines and maximum 22 lines per image, compared to *e.g.* [27], where they used 130 lines and 50 points per image) the proposed Cayley-LS solver can estimate the absolute and relative poses quite robustly, even independently of the selection of the reference camera. In 16 out of the 17 configurations all cameras have a correct pose estimated with a maximum forward projection error of less than 30 cm, except one camera, that might have had too much noise on the 3D lines. Obviously in the 17<sup>th</sup> configuration, when this camera is chosen as reference, the errors propagate over to multiple other cameras, thus only a total of 12 cameras have lower than 30 cm maximum forward projection error. In the other 16 cases median rotation errors are below  $1^\circ$  along all three axes, while median translation errors below 40 cm. Considering the distance of the cameras from the scene walls is between 10 – 25 meters, these results prove the precision and robustness of the proposed method, even for low number of lines in different multi-view setups.

**Fusion** Based on the calculated camera poses the 2D-3D fusion can be performed by coloring the scene pointcloud

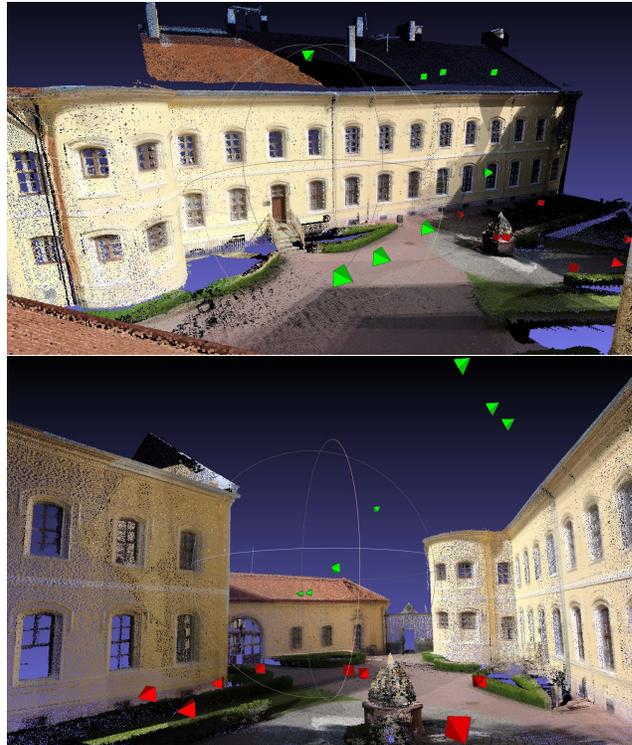


Figure 6. Fusion result shown as colored pointcloud with estimated omni (red) and perspective (green) camera positions illustrated.

from all the available cameras. An important step of this is checking the visibility of the surfaces from each given viewpoint, since in a complex urban environment occlusions can easily happen. For this purpose we used the hidden point remove tool of [14] to obtain only the visible scene-points from each camera, then we averaged out the color values proposed by multiple cameras for each vertex. The resulting colored pointcloud for the dataset can be seen in Fig. 6, including the estimated camera poses.

## 6. Conclusion

We proposed a novel robust pose estimation method for central perspective and omnidirectional cameras using line correspondences. Due to the Cayley representation of the rotation, our approach yields a low-order polynomial system both for the minimal as well as for the general  $n$ -line case, which can be efficiently solved using Grobner basis solvers. The proposed method is able to deal with outliers as well as noise on the line parameters. It compares favorably to State-of-the-Art methods, being more robust to 3D noise than AlgLS and SRPnL. The efficiency of the proposed solution was validated both on synthetic and real data with omnidirectional and perspective camera setups.

## References

- [1] H. Abdellali and Z. Kato. Absolute and relative pose estimation of a multi-view camera system using 2d-3d line pairs and vertical direction. In *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*, pages 1–8, Canberra, Australia, Dec. 2018. IEEE. 2, 3
- [2] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg. Real-time self-localization from panoramic images on mobile devices. In *Proceedings of International Symposium on Mixed and Augmented Reality*, pages 37–46, Basel, Switzerland, Oct. 2011. IEEE Computer Society. 3
- [3] S. Baker and S. K. Nayar. A Theory of Single-Viewpoint Catadioptric Image Formation. *International Journal of Computer Vision*, 35(2):175–196, 1999. 3
- [4] J. Bermudez-Cameo, G. Lopez-Nicolas, and J. J. Guerrero. Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry. *International Journal of Computer Vision*, 114(1):16–37, Aug. 2015. 2, 8
- [5] F. Camposeco, T. Sattler, and M. Pollefeys. Minimal solvers for generalized pose and scale estimation from two rays and one point. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Proceedings of European Conference Computer Vision*, volume 9909 of *Lecture Notes in Computer Science*, pages 202–218, Amsterdam, The Netherlands, Oct. 2016. Springer. 2, 3
- [6] H. Chen. Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):530–541, 1991. 3
- [7] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 1, 2, 4
- [8] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems. In *European Conference on Computer Vision*, pages 445–462, Dublin, Ireland, June 2000. 2, 3
- [9] R. Hartley, J. Trunpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, Jan 2013. 3
- [10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2004. 5
- [11] N. Horanyi and Z. Kato. Generalized pose estimation from line correspondences with known vertical direction. In *Proceedings of International Conference on 3D Vision*, pages 1–10, Qingdao, China, Oct. 2017. IEEE. 2, 3
- [12] N. Horanyi and Z. Kato. Multiview absolute pose using 3D - 2D perspective line correspondences and vertical direction. In *Proceedings of ICCV Workshop on Multiview Relationships in 3D Data*, pages 1–9, Venice, Italy, Oct. 2017. IEEE. 2
- [13] J. Kannala and S. S. Brandt. A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, 2006. 3
- [14] S. Katz, A. Tal, and R. Basri. Direct visibility of point sets. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. 8
- [15] T. Ke and S. I. Roumeliotis. An efficient algebraic solution to the perspective-three-point problem. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1–9, Honolulu, HI, USA, July 2017. IEEE. 1
- [16] L. Kneip. *Polyjam*, 2015 [online]. url:<https://github.com/laurentkneip/polyjam>. 4, 7
- [17] L. Kneip, H. Li, and Y. Seo. UPnP: an optimal  $O(n)$  solution to the absolute pose problem with universal applicability. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Proceedings of European Conference Computer Vision, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 127–142, Zurich, Switzerland, Sept. 2014. Springer. 1, 2, 3, 8
- [18] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *Proceedings of European Conference on Computer Vision*, pages 302–315. Springer Berlin Heidelberg, 2008. 4, 5, 7
- [19] V. Larsson, K. Astrom, and M. Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jul 2017. 4
- [20] G. H. Lee. A minimal solution for non-perspective pose estimation from line correspondences. In *Proceedings of European Conference on Computer Vision*, pages 170–185, Amsterdam, The Netherlands, Oct. 2016. Springer. 2, 3, 4
- [21] G. H. Lee, F. Fraundorfer, and M. Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, Portland, OR, USA, June 2013. 2, 3
- [22] G. H. Lee, M. Pollefeys, and F. Fraundorfer. Relative pose estimation for a multi-camera system with known vertical direction. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 540–547, Columbus, OH, USA, June 2014. 2
- [23] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: an accurate  $O(n)$  solution to the PnP problem. *International Journal of Computer Vision*, 81(2), 2009. 1
- [24] 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*, 34(7):1444–1450, 2012. 1, 3
- [25] B. Mičušík. *Two-View Geometry of Omnidirectional Cameras*. Phd thesis, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic, June 2004. 2
- [26] B. Mičušík and T. Pajdla. Para-catadioptric Camera Auto-calibration from Epipolar Geometry. In *Asian Conference on Computer Vision*, volume 2, pages 748–753, Seoul, Korea South, January 2004. 2
- [27] P. Miraldo, T. Dias, and S. Ramalingam. A minimal closed-form solution for multi-perspective pose estimation using points and lines. In *Computer Vision – ECCV 2018*, pages 490–507. Springer International Publishing, 2018. 2, 8

- [28] F. M. Mirzaei and S. I. Roumeliotis. Globally optimal pose estimation from line correspondences. In *International Conference on Robotics and Automation*, pages 5581–5588, Shanghai, China, May 2011. IEEE, IEEE. 1, 2
- [29] D. P. Paudel, A. Habed, C. Demonceaux, and P. Vasseur. Robust and optimal registration of image sets and structured scenes via sum-of-squares polynomials. *International Journal of Computer Vision*, 127(5):415–436, May 2019. 1
- [30] R. Pless. Using many cameras as one. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2003. 3
- [31] D. Scaramuzza, A. Martinelli, and R. Siegwart. A Toolbox for Easily Calibrating Omnidirectional Cameras. In *International Conference on Intelligent Robots*, pages 5695–5701, Beijing, October 2006. 2, 3
- [32] L. Tamas, R. Frohlich, and Z. Kato. Relative pose estimation and fusion of omnidirectional and lidar cameras. In L. de Agapito, M. M. Bronstein, and C. Rother, editors, *Proceedings of the ECCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*, volume 8926 of *Lecture Notes in Computer Science*, pages 640–651, Zurich, Switzerland, Sept. 2014. Springer. 1, 3
- [33] L. Tamas and Z. Kato. Targetless calibration of a lidar - perspective camera pair. In *Proceedings of ICCV Workshop on Big Data in 3D Computer Vision*, pages 668–675, Sydney, Australia, Dec. 2013. IEEE, IEEE. 1
- [34] C. J. Taylor and D. J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1021–1032, Nov. 1995. 3, 4
- [35] P. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, apr 2000. 4, 7
- [36] A. Vakhitov, V. Lempitsky, and Y. Zheng. Stereo relative pose from line and point feature triplets. In *Computer Vision – ECCV 2018*, pages 662–677. Springer International Publishing, 2018. 8
- [37] R. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, apr 2010. 8
- [38] P. Wang, G. Xu, and Y. Cheng. A novel algebraic solution to the perspective-three-line problem. *Computer Vision and Image Understanding*, 2018. in press. 2, 3
- [39] P. Wang, G. Xu, Y. Cheng, and Q. Yu. Camera pose estimation from lines: a fast, robust and general method. *Machine Vision and Applications*, feb 2019. 2, 3
- [40] F. Wientapper, M. Schmitt, M. Fraissinet-Tachet, and A. Kuijper. A universal, closed-form approach for absolute pose problems. *Computer Vision and Image Understanding*, 173:57–75, aug 2018. 2
- [41] F. C. Wu, Z. H. Wang, and Z. Y. Hu. Cayley transformation and numerical stability of calibration equation. *International Journal of Computer Vision*, 82(2):156–184, dec 2008. 3
- [42] C. Xu, L. Zhang, L. Cheng, and R. Koch. Pose estimation from line correspondences: A complete analysis and a series of solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1209–1222, 2017. 1, 2, 3
- [43] Y.-T. Yeh, C.-K. Peng, K.-W. Chen, Y.-S. Chen, and Y.-P. Hung. Driver assistance system providing an intuitive perspective view of vehicle surrounding. In *ACCV Workshops*, 2014. 2
- [44] B. Zhang, V. V. Appia, I. Pekkucuksen, Y. Liu, A. U. Batur, P. Shastry, S. Liu, S. Sivasankaran, and K. Chitnis. A surround view camera solution for embedded systems. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 676–681, 2014. 2
- [45] L. Zhang, C. Xu, K. Lee, and R. Koch. Robust and efficient pose estimation from line correspondences. In K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, editors, *Proceedings of Asian Conference on Computer Vision*, volume 7726 of *Lecture Notes in Computer Science*, pages 217–230, Daejeon, Korea, Nov. 2012. Springer. 2, 3
- [46] M. Zhang, Y. Chen, and G. Wang. Hybrid silhouette and key-point driven 3D-2D registration. In *Image and Graphics (ICIG), 2013 Seventh International Conference on*, pages 585–590, July 2013. 1