Globally Optimal Camera Orientation Estimation from Line Correspondences by BnB algorithm

Yinlong Liu, Guang Chen, and Alois Knoll, *Senior Member, IEEE*

***Abstract*—This letter is concerned with the problem of estimating camera orientation from a set of 2D/3D line correspondences, which is a major part of the Perspective-n-Line (PnL) problem. There are some cases that usually occur in real applications for PnL: the input line correspondences are corrupted by mismatches (a.k.a. outlier correspondences). The *RANdom SAmple Consensus* (RANSAC) algorithm is the de facto standard for solving outlier-contaminated PnL problems. However, RANSAC is a non-deterministic algorithm, which means that it produces a reasonable result only with a certain probability. Therefore, a PnL algorithm that could obtain a certifiably optimal solution from outlier-contaminated data is a matter of priority for some safety-critical applications. In this letter, we take a big step towards this goal by investigating globally optimal camera orientation estimation algorithms. Firstly, we decouple the rotation and translation estimation of a PnL problem by considering the geometrical property of the PnL problem. The Branch-and-Bound (BnB) algorithm is applied and it globally searches the entire rotation space to obtain the optimal camera orientation. To investigate the performance of our method, we tested the proposed algorithm on both synthetic and real data, and the results show that our algorithms can obtain the optimal camera orientation and are more robust than several state-of-the-art PnL methods.**

***Index Terms*—Vision-based navigation, SLAM.**

I. Introduction

ABSOLUTE camera pose estimation is determining the position and orientation of a camera in the scene, which is a core task in computer vision and robot navigation [1]. This task can be solved using 2D/3D line feature correspondences, which is also known as the Perspective-n-Line (PnL) problem [2]. It plays an import role in many computer vision applications, e.g., Simultaneous Localization and Mapping (SLAM) [3] and robot localization and navigation [4], [5]. It is worth noting that the PnL approach is inherently suitable for texture-less scenes, e.g., man-made structural environments [6], [7].

Manuscript received July 18, 2020; accepted October 29, 2020. Date of publication November 16, 2020; date of current version November 27, 2020. This letter was recommended for publication by Associate Editor P. Vasseur and Editor Eric Marchand upon evaluation of the Reviewers' comments. This work was supported in part by the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement 945539 (Human Brain Project SGA3), in part by the National Natural Science Foundation of China under Grant 61906138, and in part by the Shanghai AI Innovative Development Project 2018. The work of Yinlong Liu was supported by Chinese Scholarship Council(CSC). *(Corresponding author: Guang Chen.)*

Yinlong Liu and Alois Knoll are with the Department of Informatics, Technische Universität München, München 85748, Germany (e-mail: Yinlong.liu@tum.de; knoll@in.tum.de).

Guang Chen is with the Tongji University, Shanghai 200092, China, and also with the Technische Universität München, München 80333, Germany (e-mail: guang@in.tum.de).

Digital Object Identifier 10.1109/LRA.2020.3037843

To address the PnL problem, one of the most important preconditions is knowing the correspondences between 3D line features in real world and their 2D counterparts in the image plane. Unfortunately, incorrect correspondences, which are also known as outliers, are usually unavoidable in the real applications. These mismatches seriously impair the camera pose estimation: even a small percentage of outliers might significantly decrease the accuracy of outlier-free algorithms [8]. To reduce the impact of corrupted data, the de facto standard mechanism is combining an outlier-free PnL algorithm with a *RANdom Sample Consensus* (RANSAC) scheme [6], [9]. Broadly speaking, RANSAC randomly samples minimal subsets of the inputs and applies the embedded outlier-free PnL algorithm to obtain candidate solutions. After repeating the sampling routine many times, RANSAC returns the best candidate with the largest inlier set as the final solution, which is also known as a hypothesize-and-verify framework. However, the randomized nature of RANSAC does provide no guarantee of the optimality of its solution. In other words, there is no absolute certainty that the obtained result is a satisfactory solution [10].

Nevertheless, there are some safety-critical applications (e.g., self-driving cars) which demand that such spatial perception algorithms can provide certifiably optimal solutions in the presence of noise and outliers [11], [12]. Conventional methods that combine outlier-free PnL algorithms and RANSAC fail to achieve this goal. In this letter, we take a big step towards this objective: we propose a novel method for obtaining the globally optimal camera orientation from outlier-contaminated line correspondences. Furthermore, as the camera pose comprises position (i.e., translation $\boldsymbol{t} \in \mathbb{R}^3$) and orientation (i.e., rotation $\boldsymbol{R} \in SO(3)$), obtaining the provably optimal camera orientation is usually favorable for addressing the problem of the certifiably optimal camera pose [13], [14].

In addition, many modern vision systems are equipped with Inertial Measurement Units (IMUs), which could provide a prior knowledge of the vertical direction [5], [15]. Accordingly, we first proposed a singularity-free and thus more accurate algorithm for the outlier-free known-vertical-direction PnL problem. Furthermore, for outlier-contaminated cases, a special camera orientation estimation algorithm with a guarantee of the optimality is proposed.

The rest of the letter is organized as follows: Prior arts are reviewed in Sect. II. The proposed methods are described in Sect. III, IV and V. To verify the feasibility of the proposed methods, synthetic and real-world experiments are conducted in Sect. VI. Lastly, we conclude the work in Sect. VII.

2377-3766 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on July 27,2022 at 09:52:32 UTC from IEEE Xplore. Restrictions apply.

II. Related Work

Conventionally, combing outlier-free PnL algorithms with RANSAC is one of the most commonly used mechanisms for estimating the camera pose from outlier-contaminated line correspondences. For the RANSAC scheme [16], some recent advancements focus on special geometrical problems (e.g., pseudo-convex [17], [18]); other recent advancements still do not change its randomized nature and still lack absolute certainty that the obtained solution is optimal [19], [20]. In any case, they are not directly used to address PnL problems. Therefore, we review the most relevant PnL algorithms in detail.

1) PnL Algorithms: According to the optimizing techniques, we can divide these methods into three categories:

a) Locally iterative PnL solutions [21]–[23]. Generally speaking, these local iterative algorithms formulate the pose estimation problem as a nonlinear least squares optimization problem. Unfortunately, the objectives usually are non-convex because of noise and outliers. Therefore, if the initialization is not carefully set, these locally iterative algorithms might be trapped in a local optimum, which might be far from the true camera pose [6].

b) Algebraic solutions [24]–[26]. The algebraic algorithms estimate the camera pose by solving a polynomial system of equations. One of the most important advantages of these algebraic algorithms is that they can obtain the globally optimal solution from outlier-free inputs without careful initializing. However, to obtain a robust solution from outlier-contaminated inputs, they must be nested inside a non-deterministic RANSAC framework.

c) Linearized PnL solutions [2], [6]. The linearized PnL methods formulate the line correspondences as a homogeneous system of linear equation by dropping some constraints that might compromise the accuracy [15], [26]. The final camera pose can be extracted from the solution of the linear system by adding the dropped constraints.

It is worth noting that to suppress outlier inputs, these linearized PnL methods can incorporate with Algebraic Outlier Rejection (AOR [27]) [6]. AOR can estimate the robust solution of the linear objective very quickly. Therefore, the camera pose can be recovered from the robust linear solution. However, these methods usually have a *break-point*, which means that when the proportion of outliers reaches the *break-point*, these methods cannot obtain a satisfactory solution [6], [27].

2) Known-Vertical-Direction PnL Algorithms: With the help of relatively cheap IMUs, some recent work has focused on estimating camera pose from line correspondences with a known vertical direction [5], [15], [28]. In common, they first estimate the camera orientation and then solve translation. Specially, linearized algorithms are proposed for estimating camera rotation in [15] and [5]. The authors in [15] pointed out that the biggest disadvantage of a linear solution is ignoring orthogonality, which might lead to low accuracy.

To consider the orthogonality of camera rotation, a cubic polynomial solution, which belongs to the algebraic algorithm, is proposed for determining the camera orientation [28], [29]. However, it employs a singularity-affected rotation parameterization that will reduce the accuracy in some cases. In this letter, we apply a singularity-free rotation parameterization to improve accuracy.

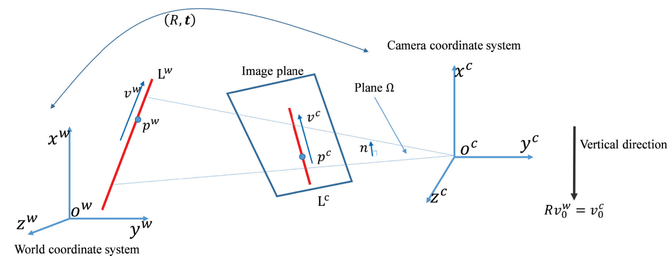

Fig. 1. Line definition and the geometrical constraints in PnL problem.

For outlier-contaminated inputs, a novel RANSAC2 algorithm, whose inner routine requires only two random 2D-3D pairs of line correspondences, is proposed to estimate the rotation [5]. Actually, one line correspondence is sufficient to be nested inside RANSAC [30]. Nonetheless, RANSAC still cannot guarantee the optimality of the solution.

III. Rotation Estimation for PnL

A. Mathematical Formulation

Given a calibrated camera, the lines in the image plane can be denoted as L^c in the camera coordinate system $\{\boldsymbol{o}^c\boldsymbol{x}^c\boldsymbol{y}^c\boldsymbol{z}^c\}$ and the lines in the real world can be denoted as L^w in the world coordinate system $\{\boldsymbol{o}^w\boldsymbol{x}^w\boldsymbol{y}^w\boldsymbol{z}^w\}$ (see Fig. 1). Geometrically, a 3D line can be represented by a point $\boldsymbol{p} \in \mathbb{R}^3$ on the line and a unit direction $\boldsymbol{v} \in \mathbb{R}^3$ denoting the direction of the line, therefore L^c and L^w can be represented as: $L^c = \boldsymbol{p}^c + \lambda_c \boldsymbol{v}^c$ and $L^w = \boldsymbol{p}^w + \lambda_w \boldsymbol{v}^w$, where λ_c and λ_w are parameters describing a particular point on each line. Note that the point $\boldsymbol{p}^c$ is not necessarily the corresponding image point of the 3D point $\boldsymbol{p}^w$. The objective of PnL is to estimate the camera pose (i.e., rotation/orientation $\boldsymbol{R}$ and translation/position $\boldsymbol{t}$) from a set of line correspondences $\{(L_i^w, L_i^c)\}_{i=1}^N$.

To obtain globally optimal rotation, we first formulate constraints about rotation $\boldsymbol{R}$ by constructing an auxiliary variable. For a line correspondence, there is a projection plane Ω passing through the camera center and both lines. The unit normal of the plane Ω is denoted by $\boldsymbol{n}$ in the camera coordinate system and it can be obtained from the line L^c in the image plane: $\boldsymbol{n} = (\boldsymbol{v}^c \times \boldsymbol{p}^c)/\|\boldsymbol{v}^c \times \boldsymbol{p}^c\|$. For the i-th line pair, there is an important geometrical constraint [15], [31].

$$\boldsymbol{n}_i^T \boldsymbol{R} \boldsymbol{v}_i^w = 0 \tag{1}$$

Furthermore, there is another constraint for translation $\boldsymbol{t}$.

$$\boldsymbol{n}_i^T (\boldsymbol{R} \boldsymbol{p}_i^w + \boldsymbol{t}) = 0; \tag{2}$$

Conventionally, the outlier-free PnL algorithms formulate the objective as [26], [31].

$$\min_{R} \sum_{i=1}^{N} \left(\boldsymbol{n}_i^T \boldsymbol{R} \boldsymbol{v}_i^w\right)^2 \tag{3a}$$

$$\min_{\boldsymbol{t}} \sum_{i=1}^{N} \left(\boldsymbol{n}_i^T (\boldsymbol{R} \boldsymbol{p}_i^w + \boldsymbol{t})\right)^2 \tag{3b}$$

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on July 27,2022 at 09:52:32 UTC from IEEE Xplore. Restrictions apply.

Camera pose can be obtained by optimizing these two objectives sequentially. However, it is well known that the objectives are sensitive to outliers [2], [6]. In contrast to the outlier-free PnL algorithm, our robust objective is formulated by maximizing the cardinality of the inlier set [32].

$$\max_{\boldsymbol{R}} \sum_{i=1}^{N} \mathbb{I}\left(|\boldsymbol{n}_i^T \boldsymbol{R} \boldsymbol{v}_i^w| \leq \varepsilon\right) \tag{4a}$$

$$\max_{\boldsymbol{t}} \sum_{i=1}^{N} \mathbb{I}\left(|\boldsymbol{n}_i^T (\boldsymbol{R} \boldsymbol{p}_i^w + \boldsymbol{t})| \leq \varepsilon\right) \tag{4b}$$

where ε is the inlier threshold and $\mathbb{I}(\cdot)$ is the indicator function. Specifically, $\mathbb{I}(\cdot)$ returns 1 if the condition $\cdot$ is true, and it returns 0 if the condition $\cdot$ is not true. The maximization cardinality formulation is inherently robust to outliers and is successfully applied in many applications of robust estimation [32].

B. Branch-and-Bound

We focus on obtaining the globally optimal solution of camera orientation (i.e., solving Eq. (4a)). To obtain the globally optimal solution, we introduce the Branch-and-Bound algorithm (BnB) [33], which is a global optimization technique that is applied in many geometrical vision problems (e.g., 3D point cloud registration [34], [35]).

Specifically, the BnB algorithm recursively divides the best possible branch (i.e., solution domain) into small branches (i.e., branching), then it calculates the upper and lower bounds of the optimum in each divided branches (i.e., bounding). By checking the upper and lower estimated bounds, it removes some divided branch which cannot produce a better solution than the best one found by the algorithm so far (i.e., pruning). The branching-bounding-pruning process is repeated until the desired accuracy is approached and then the optimal solution is found.

In this letter, the minimally parameterized axis-angle parameterization is used to represent the rotation space [35]. Specifically, a 3D rotation $\boldsymbol{R} \in SO(3)$ corresponds to a 3-vector $\boldsymbol{r} \in \mathbb{R}^3$ whose direction and norm specify the axis and angle of rotation. Their relationship follows Rodrigues' rotation formula:

$$\boldsymbol{R} = \exp([\boldsymbol{r}]_\times) = \mathrm{I} + \sin(\theta)[\hat{\boldsymbol{r}}]_\times + (1 - \cos(\theta))\,[\hat{\boldsymbol{r}}]_\times^2 \tag{5}$$

where I is the 3×3 identity matrix; $\theta = ||\boldsymbol{r}||$ is the angle of the rotation; $\hat{\boldsymbol{r}} = \boldsymbol{r}/||\boldsymbol{r}||$ is the axis of the rotation; $[\bullet]_\times$ is the skew-symmetric matrix for a vector $\bullet$ and $\exp(\cdot)$ is the matrix exponential of the $SO(3)$ algebra [36]. With the help of axis-angle parameterization, all possible rotations are in a π-ball [35]. For ease of manipulation, a cube enclosing the π-ball is used as the initial searching domain. Therefore, the branching process in the BnB algorithm divides the larger cube into eight sub-cubes.

The key part of the BnB algorithm is estimating the upper and lower bounds efficiently and tightly in each divided branch. Given a divided cube-shape rotation branch, whose center is $\boldsymbol{r}_0$, the lower bound of the optimum can be set as

$$e_L = \sum_{i=1}^{N} \mathbb{I}\left(|\boldsymbol{n}_i^T \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w| \leq \varepsilon\right) \tag{6}$$

where $\boldsymbol{R}_{r_0}$ is the matrix form of rotation $\boldsymbol{r}_0$.

Proof: The function value at a specific point (i.e. $\boldsymbol{R}_{r_0}$) within the domain must be less than the maximum value. ■

On the other hand, given a divided cube-shape rotation branch, whose center is $\boldsymbol{r}_0$ and whose side length is ϕ, the upper bound of the optimum can be set as

$$e_U = \sum_{i=1}^{N} \mathbb{I}\left(|\boldsymbol{n}_i^T R_{r_0} \boldsymbol{v}_i^w| \leq \cos\left(\lfloor \arccos(\varepsilon) - \psi \rfloor\right)\right) \tag{7}$$

where $\boldsymbol{R}_{r_0}$ is the matrix form of rotation $\boldsymbol{r}_0$; $\psi \triangleq \min\{\sqrt{3}\phi/2, \pi\}$; $\lfloor \cdot \rfloor$ is a non-negative function, which means if $\theta < 0$, $\lfloor \theta \rfloor = 0$ and if $\theta \geq 0$, $\lfloor \theta \rfloor = \theta$.

Proof: To derive the upper bound, we first introduce a famous lemma from [35]:

Lemma 1: For an arbitrary vector $\boldsymbol{v} \in \mathbb{R}^3$ and two rotations $\boldsymbol{R}_r$ and $\boldsymbol{R}_{r_0}$ in matrix form, $\boldsymbol{r}$ and $\boldsymbol{r}_0$ in angle-axis form, then

$$\angle(\boldsymbol{R}_r \boldsymbol{v}, \boldsymbol{R}_{r_0} \boldsymbol{v}) \leq ||\boldsymbol{r} - \boldsymbol{r}_0|| \tag{8}$$

where $\angle(\cdot, \cdot)$ is the angle of two vectors.

According to *Lemma* 1, given a divided cube-shape rotation branch, whose center is $\boldsymbol{r}_0$ and side is ϕ, we have

$$\angle(\boldsymbol{R}_r \boldsymbol{v}_i^w, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w) \leq \min\{||\boldsymbol{r} - \boldsymbol{r}_0||, \pi\} \tag{9}$$

$$\leq \min\left\{\frac{\sqrt{3}}{2}\phi, \pi\right\} \triangleq \psi \tag{10}$$

where $\boldsymbol{r}$ is an arbitrary point in the cube, whose matrix form is $\boldsymbol{R}_r$. According to triangle inequality

$$\angle\left(\boldsymbol{n}_i, \boldsymbol{R}_r \boldsymbol{v}_i^w\right) \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) + \angle\left(\boldsymbol{R}_r \boldsymbol{v}_i^w, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) \tag{11}$$

$$\leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) + \psi \tag{12}$$

$$\angle\left(\boldsymbol{n}_i, \boldsymbol{R}_r \boldsymbol{v}_i^w\right) \geq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) - \angle\left(\boldsymbol{R}_r \boldsymbol{v}_i^w, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) \tag{13}$$

$$\geq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) - \psi \tag{14}$$

Hence,

$$\angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) - \psi \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_r \boldsymbol{v}_i^w\right) \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) + \psi \tag{15}$$

Observe

$$\mathbb{I}\left(|\boldsymbol{n}_i^T \boldsymbol{R}_r \boldsymbol{v}_i^w| \leq \varepsilon\right) \tag{16}$$

$$\Leftrightarrow \mathbb{I}\left(-\varepsilon \leq \boldsymbol{n}_i^T \boldsymbol{R}_r \boldsymbol{v}_i^w \leq \varepsilon\right) \tag{17}$$

$$\Leftrightarrow \mathbb{I}\left(-\varepsilon \leq \cos\left(\angle\left(\boldsymbol{n}_i, \boldsymbol{R}_r \boldsymbol{v}_i^w\right)\right) \leq \varepsilon\right) \tag{18}$$

$$\Leftrightarrow \mathbb{I}\left(\arccos(\varepsilon) \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_r \boldsymbol{v}_i^w\right) \leq \pi - \arccos(\varepsilon)\right) \tag{19}$$

Also, we note that

$$\arccos(\varepsilon) \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_r \boldsymbol{v}_i^w\right) \leq \pi - \arccos(\varepsilon) \tag{20}$$

$$\Rightarrow \begin{cases} \arccos(\varepsilon) & \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_r \boldsymbol{v}_i^w\right) \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) + \psi \\ \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) - \psi & \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_r \boldsymbol{v}_i^w\right) \leq \pi - \arccos(\varepsilon) \end{cases} \tag{21}$$

$$\Rightarrow \begin{cases} \arccos(\varepsilon) & \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) + \psi \\ \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) - \psi & \leq \pi - \arccos(\varepsilon) \end{cases} \tag{22}$$

$$\Rightarrow \arccos(\varepsilon) - \psi \leq \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0} \boldsymbol{v}_i^w\right) \leq \pi - \arccos(\varepsilon) + \psi \tag{23}$$

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on July 27,2022 at 09:52:32 UTC from IEEE Xplore. Restrictions apply.

$$\Rightarrow \lfloor \arccos(\varepsilon) - \psi \rfloor \le \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0}\boldsymbol{v}_i^w\right) \le \pi - \lfloor \arccos(\varepsilon) - \psi \rfloor \tag{24}$$

$$\Rightarrow \begin{array}{l} -\cos(\lfloor \arccos(\varepsilon) - \psi \rfloor) \le \cos\left(\angle\left(\boldsymbol{n}_i, \boldsymbol{R}_{r_0}\boldsymbol{v}_i^w\right)\right) \\ \qquad \le \cos(\lfloor \arccos(\varepsilon) - \psi \rfloor) \end{array} \tag{25}$$

$$\Rightarrow \begin{array}{l} -\cos(\lfloor \arccos(\varepsilon) - \psi \rfloor) \le \boldsymbol{n}_i^T \boldsymbol{R}_{r_0}\boldsymbol{v}_i^w \\ \qquad \le \cos(\lfloor \arccos(\varepsilon) - \psi \rfloor) \end{array} \tag{26}$$

$$\Rightarrow |\boldsymbol{n}_i^T \boldsymbol{R}_{r_0}\boldsymbol{v}_i^w| \le \cos\left(\lfloor \arccos(\varepsilon) - \psi \rfloor\right) \tag{27}$$

where Eq. (24) follows $0 \le \angle(\boldsymbol{n}_i, \boldsymbol{R}_{r_0}\boldsymbol{v}_i^w) \le \pi$.

Therefore,

$$|\boldsymbol{n}_i^T \boldsymbol{R}_r \boldsymbol{v}_i^w| \le \varepsilon \tag{28}$$

$$\Leftrightarrow \arccos(\varepsilon) \le \angle\left(\boldsymbol{n}_i, \boldsymbol{R}_r\boldsymbol{v}_i^w\right) \le \pi - \arccos(\varepsilon) \tag{29}$$

$$\Rightarrow |\boldsymbol{n}_i^T R_{r_0}\boldsymbol{v}_i^w| \le \cos\left(\lfloor \arccos(\varepsilon) - \psi \rfloor\right) \tag{30}$$

Hence,

$$\mathbb{I}\left(|\boldsymbol{n}_i^T \boldsymbol{R}_r \boldsymbol{v}_i^w| \le \varepsilon\right) \le \mathbb{I}\left(|\boldsymbol{n}_i^T \boldsymbol{R}_{r_0}\boldsymbol{v}_i^w| \le \cos\left(\lfloor \arccos(\varepsilon) - \psi \rfloor\right)\right) \tag{31}$$

And,

$$\max \sum_{i=1}^{N} \mathbb{I}\left(|\boldsymbol{n}_i^T \boldsymbol{R}\boldsymbol{v}_i^w| < \varepsilon\right) \le e_U \tag{32}$$

Therefore, e_U is an upper bound

IV. Rotation Estimation for Known-Vertical PnL

A. Mathematical Formulation

In this section, we consider the special case that the vertical direction is known. Geometrically, prior knowledge of the vertical direction is a constraint [15]: $R\boldsymbol{v}_0^w = \boldsymbol{v}_0^c$, where $\boldsymbol{v}_0^w$ and $\boldsymbol{v}_0^c$ are the unit-norm vertical direction in the world coordinate system and camera coordinate system, respectively. The solution of $R\boldsymbol{v}_0^w = \boldsymbol{v}_0^c$ is $R = R_{\boldsymbol{v}_0^c} \cdot R_w^c$, where R_w^c is the rotation that aligns $\boldsymbol{v}_0^w$ to $\boldsymbol{v}_0^c$ with the minimum geodesic motion (see [37]); $R_{\boldsymbol{v}_0^c} = \exp(\alpha[\boldsymbol{v}_0^c]_\times)$ is a rotation that rotates α about axis $\boldsymbol{v}_0^c$. Hence, the objective of our work is to determine α correctly.

For the i-th line pair constraints:

$$\boldsymbol{n}_i^T R \boldsymbol{v}_i^w \tag{33}$$

$$= \boldsymbol{n}_i^T \cdot R_{\boldsymbol{v}_0^c} \cdot R_w^c \boldsymbol{v}_i^w \tag{34}$$

$$= \boldsymbol{n}_i^T \cdot \left(\mathrm{I} + \sin(\alpha)[\boldsymbol{v}_0^c]_\times + (1 - \cos(\alpha))\,[\boldsymbol{v}_0^c]_\times^2\right) \cdot R_w^c \boldsymbol{v}_i^w \tag{35}$$

$$\begin{aligned} &= (-\boldsymbol{n}_i^T \cdot [\boldsymbol{v}_0^c]_\times^2 \cdot R_w^c \boldsymbol{v}_i^w) \cdot \cos(\alpha) + (\boldsymbol{n}_i^T \cdot [\boldsymbol{v}_0^c]_\times \cdot R_w^c \boldsymbol{v}_i^w) \cdot \sin(\alpha) \\ &\quad + \boldsymbol{n}_i^T \cdot \left(\mathrm{I} + [\boldsymbol{v}_0^c]_\times^2\right) \cdot R_w^c \boldsymbol{v}_i^w \end{aligned} \tag{36}$$

$$= a_i \cdot \cos(\alpha) + b_i \cdot \sin(\alpha) + c_i \tag{37}$$

where $a_i = -\boldsymbol{n}_i^T \cdot [\boldsymbol{v}_0^c]_\times^2 \cdot R_w^c \boldsymbol{v}_i^w$; $b_i = \boldsymbol{n}_i^T \cdot [\boldsymbol{v}_0^c]_\times \cdot R_w^c \boldsymbol{v}_i^w$; $c_i = \boldsymbol{n}_i^T \cdot (\mathrm{I} + [\boldsymbol{v}_0^c]_\times^2) \cdot R_w^c \boldsymbol{v}_i^w$. Therefore, given the known vertical direction, the rotation constraint from the i-th line correspondence is:

$$\boldsymbol{n}_i^T R \boldsymbol{v}_i^w = a_i \cdot \cos(\alpha) + b_i \cdot \sin(\alpha) + c_i = 0 \tag{38}$$

B. Unit Constraint Solution

To address the unknown α, a linear solution is proposed in [15] and [5], where $q_1 = \cos(\alpha)$, $q_2 = \sin(\alpha)$, then for the i-th line correspondence:

$$a_i \cdot q_1 + b_i \cdot q_2 + c_i = 0 \tag{39}$$

Hence, q_1 and q_2 can be determined from the linear system. However, they may not satisfy the trigonometric constraint $q_1^2 + q_2^2 = 1$ [5], [15]. In addition, the cubic polynomial solution is proposed in [15], [28], [29]. Let $q = \tan(\alpha/2)$, then, $\cos(\alpha) = (1 - q^2)/(1 + q^2)$ and $\sin(\alpha) = 2q/(1 + q^2)$. Therefore, for the i-th line correspondence:

$$a_i \cdot (1 - q^2)/(1 + q^2) + b_i \cdot 2q/(1 + q^2) + c_i = 0 \tag{40}$$

The objective is $\min_q \sum_{i=1}^N ((c_i - a_i)q^2 + 2b_i q + a_i + c_i)^2$. Hence, q can be solved from the polynomial system. However, when $\alpha = \pi, q \to +\infty$ and when $\alpha = -\pi, q \to -\infty$. This is a singularity-affected parameterization [38], which might reduce the accuracy near the singularity angle (i.e., $\pm\pi$).

In this section, we apply a singularity-free parameterization to increase the accuracy. Specifically, the objective is formulated as:

$$\min \sum_{i=1}^{N} (a_i \cdot q_1 + b_i \cdot q_2 + c_i)^2, \qquad \text{s.t. } q_1^2 + q_2^2 = 1 \tag{41}$$

In contrast to the linear solution, it does not drop the unit-norm constraint. Additionally, in contrast to the cubic polynomial solution, it does not suffer from any degeneracy.

Clearly, Eq. (41) is a typical *Equality Constrained Optimization* problem [39]. The Lagrangian formulation is

$$f = \sum_{i=1}^{N} (a_i \cdot q_1 + b_i \cdot q_2 + c_i)^2 + \lambda(q_1^2 + q_2^2 - 1) \tag{42}$$

where λ is a Lagrange multiplier. The first-order optimality condition of the Lagrangian formulation is

$$\begin{cases} f'_{q_1} = 2\sum_{i=1}^N a_i^2 q_1 + 2\sum_{i=1}^N a_i b_i q_2 + 2\sum_{i=1}^N a_i c_i + 2\lambda q_1 = 0 \\ f'_{q_2} = 2\sum_{i=1}^N b_i^2 q_2 + 2\sum_{i=1}^N a_i b_i q_1 + 2\sum_{i=1}^N b_i c_i + 2\lambda q_2 = 0 \\ f'_\lambda = q_1^2 + q_2^2 - 1 = 0 \end{cases} \tag{43}$$

$$\Rightarrow \begin{cases} \widetilde{a} q_1^2 + \widetilde{b} q_2^2 + \widetilde{c} q_1 q_2 + \widetilde{d} q_1 + \widetilde{e} q_2 = 0 \\ q_1^2 + q_2^2 - 1 = 0 \end{cases} \tag{44}$$

where $\widetilde{a} = \sum_{i=1}^N a_i b_i$; $\widetilde{b} = -\sum_{i=1}^N a_i b_i$; $\widetilde{c} = \sum_{i=1}^N (b_i^2 - a_i^2)$; $\widetilde{d} = \sum_{i=1}^N b_i c_i$; $\widetilde{e} = -\sum_{i=1}^N a_i c_i$. The original optimal solution (q_1, q_2) can be obtained by identifying all solutions of Eq. (44).

C. Outlier-Contaminated Cases

In the outlier-contaminated cases, just one line correspondence is sufficient to be a minimal-subset solver in RANSAC [30]. For the i-th line correspondence, if the unit norm

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on July 27,2022 at 09:52:32 UTC from IEEE Xplore. Restrictions apply.

constraint is not ignored, then we have

$$\begin{matrix} a_i \cdot q_1 + b_i \cdot q_2 + c_i = 0 \\ q_1^2 + q_2^2 = 1 \end{matrix} \Rightarrow (a_i^2+b_i^2)q_1^2+2a_ic_iq_1+c_i^2-b_i^2=0 \tag{45}$$

We can obtain two solutions for α from Eq. (45). Similarly, two α solutions can also be obtained from Eq. (40) (singularity-affected parameterization).

As discussed above, RANSAC-type algorithms cannot guarantee the optimality of the solution. To obtain the certifiably optimal solution, we apply the BnB algorithm to obtain the optimal α from outlier-contaminated inputs. According to Eq. (4a) and Eq. (38), the robust objective can be formulated as

$$\max_{\alpha} \sum_{i=1}^{N} \mathbb{I}\left(|a_i \cdot \cos(\alpha) + b_i \cdot \sin(\alpha) + c_i| \le \varepsilon\right) \tag{46}$$

Equivalently,

$$\max_{\alpha} \sum_{i=1}^{N} \mathbb{I}\left(\left|\sqrt{a_i^2+b_i^2} \cdot \sin(\alpha + \varphi_i) + c_i\right| \le \varepsilon\right) \tag{47}$$

where $\varphi_i = \arctan 2(a_i, b_i)$.[1]

Given a branch $[\underline{\alpha}, \overline{\alpha}]$, the upper bound and lower bound can be

$$\widetilde{e}_L = \sum_{i=1}^{N} \mathbb{I}\left(\left|\sqrt{a_i^2+b_i^2} \cdot \sin(\hat{\alpha} + \varphi_i) + c_i\right| \le \varepsilon\right) \tag{48}$$

$$\widetilde{e}_U = \sum_{i=1}^{N} \mathbb{I}\left(\left|\sqrt{a_i^2+b_i^2} \cdot \sin(\alpha_i^* + \varphi_i) + c_i\right| \le \varepsilon\right) \tag{49}$$

where $\hat{\alpha} = 0.5(\underline{\alpha} + \overline{\alpha})$ and by *Interval Analysis* (see [40]) $\alpha_i^* = \arg\max_\alpha \mathbb{I}(\left|\sqrt{a_i^2+b_i^2} \cdot \sin(\alpha + \varphi_i) + c_i\right| \le \varepsilon)$

V. Translation Estimation

After obtaining the optimal camera rotation $\boldsymbol{R}$, solving the translation then becomes a linear model fitting problem [32]. Theoretically, the globally optimal translation can also be obtained by solving Eq. (4b) using the BnB algorithm [41]. However, the translation domain, which is different from the closed $SO(3)$ structure, is not easily estimated correctly for various applications. Moreover, sequentially solving rotation and translation by two separate BnB methods does not mean that the obtained optimal rotation and translation are necessarily globally optimal for the combined problem [14]. In this letter, the RANSAC algorithm[2] is applied to estimate translation.

VI. Experiments

To investigate the performance of our algorithms, we compared them with several state-of-the-art PnL methods on both synthetic and real-world data. All experiments were run on a personal computer with an AMD Ryzen 7 2700X CPU and 32GB RAM.

A. Experimental Setup

All comparison methods are listed:

- Ansar+MLESAC4+RPnL: Ansar PnL algorithm [25] is nested into MLESAC [42] and the final solution is computed by RPnL [43].
- Mirzaei+MLESAC3: Mirzaei PnL algorithm [31] is nested inside MLESAC.
- RPnL+MLESAC4: RPnL algorithm [43] is nested inside MLESAC.
- P3L+RANSAC3: A P3L algorithm [2] is plugged into RANSAC.
- ASPnL+RANSAC4: ASPnLalgorithm [2] is plugged into RANSAC.
- DLT-Lines+AOR: DLT(Direct Linear Transformation)-Lines PnL algorithm [6].
- LPnL_Bar_LS+AOR: As proposed in [2], where the lines are parameterized with the barycentric coordinates.
- LPnL_Bar_ENull+AOR: As proposed by Xu et at. in [2].
- DLT-Pücker-Lines+AOR: It applies Pücker coordinates [6].
- DLT-Combined-Lines+AOR: It is a combination of DLT-Lines and DLT-Plücker-Lines [6].
- Ro_PnL: Our proposed Robust PnL algorithm, which first applies the BnB algorithm to obtain rotation then uses RANSAC to estimate translation.
- Cubic-solution: This is an outlier-free known-vertical-direction PnL algorithm in [15] and cubic-ransac is its outlier-robust version.
- Unit-solution: Our proposed singularity-free non-minimal PnL algorithm with a known vertical direction. Unit-ransac is its outlier-robust version, which is also proposed in [30].
- vBnB: This method first estimates rotation with a known vertical direction using our proposed non-RANSAC globally optimal algorithm, then estimates translation by RANSAC.

The number at the end of MLESAC/RANSAC denotes the number of line correspondences used to generate hypotheses and the maximal number of random trials is limited to 10,000. +AOR means the linear algorithm is combined with AOR [6], [27]. The inlier threshold ε was set to 1° in RANSAC-type methods and BnB-based methods. All the compared codes[3] and our codes were executed on Matlab2019a. To demonstrate the accuracy and robustness, the translation/position error is defined as: $e_{trans} = ||\boldsymbol{t}_{gt} - \boldsymbol{t}^*||$ where $\boldsymbol{t}_{gt}$ is ground truth; $\boldsymbol{t}^*$ is the estimated translation. The rotation/orientation error is defined as the angle between ground truth $\boldsymbol{R}_{gt}$ and estimated rotation $\boldsymbol{R}^*$: $e_{rot} = \arccos(0.5 * (trace(\boldsymbol{R}_{gt}^{-1}\boldsymbol{R}^*) - 1))$. To compare the efficiency, the median runtime for 500 trials is recorded for each setting.

[1] arctan2(·, ·) is the four-quadrant inverse tangent function: https://www.mathworks.com/help/matlab/ref/atan2.html

[2] It was implemented by a Matlab built-in function: https://www.mathworks.com/help/vision/ref/ransac.html

[3] https://www.fit.vutbr.cz/∼ipribyl/DLT-based-PnL/

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on July 27,2022 at 09:52:32 UTC from IEEE Xplore. Restrictions apply.

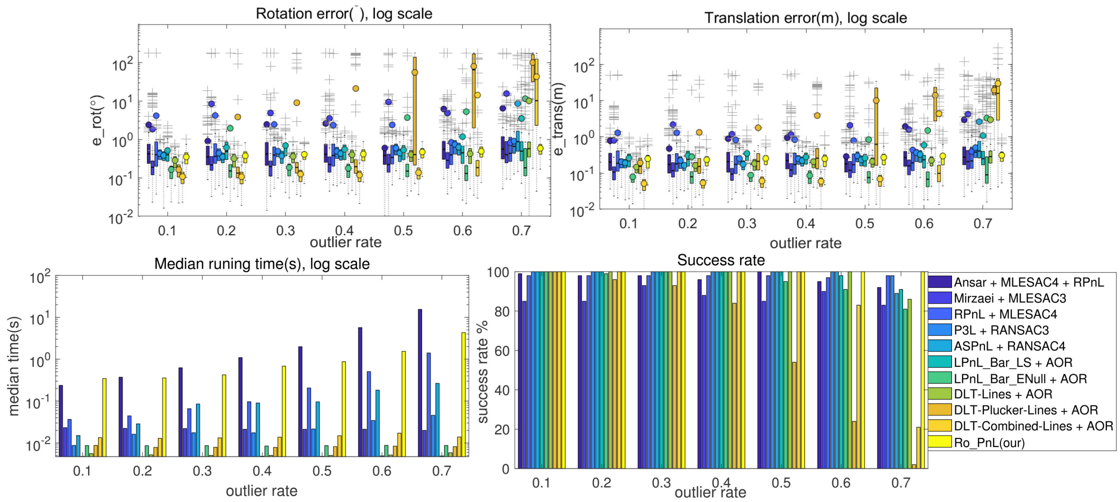

Fig. 2. Controlled experiments on synthetic data with different outlier rates.

B. Synthetic Data Experiments

$N = 200$ line segments were created randomly using $2N$ random endpoints, which were distributed in a cube $[100\,\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}]^3$. A virtual pinhole camera was randomly placed in the scene facing towards to the line segments. The camera is simulated using a 640×480 pixels image, 800 pixels focal length. To simulate noise, the 2D endpoints were perturbed with Gaussian noise with a standard deviation of $\sigma = 1$ pixel. The outliers were simulated by adding Gaussian noise with a very large standard deviation ($\sigma = 100$ pixels).

Outlier-Contaminated Inputs: We first tested our proposed method on synthetic data with different outlier rates. The outlier rates are $\{0.1, \ldots, 0.7\}$. To demonstrate the guaranteed optimality of our method, we define the success rate: N^+/N, where N is the total number of inputs; N^+ is the number of successful cases that satisfy $e_{rot} < 5^\circ$ and $e_{trans} < 2\,\mathrm{m}$. The results are shown in Fig. 2. We can draw the following two conclusions from the results: (1) With the outlier rate increasing, the existing methods might return more unsatisfactory solutions. Conversely, our proposed methods (i.e., Ro_PnL) can obtain the globally optimal camera rotation, which was superior to other existing PnL methods. In addition, obtaining optimal rotation is helpful in estimating camera translation. Therefore, our method can always provide the maximum success rate. (2) Although our proposed method returns very robust solutions, it had a longer running time than most existing methods except for the Ansar algorithm [25], which has $O(N^2)$ computational complexity.

Outlier-Free Inputs with a Known Vertical Direction: First we present the comparison between unit-solution and cubic-solution to confirm that our proposed singularity-free parameterization leads to superior accuracy. Different camera orientations ($\alpha = \{-\pi, \cdots, \pi\}$) and random orientations were also tested. The results are shown in Fig. 3. When the camera orientation is near to the singularity angle (i.e., $\pm\pi$), our proposed unit-solution was clearly able to achieve better results, and therefore, our proposed method could obtain superior accuracy in random cases. In addition, we also compared these two methods at different noise levels ($\{1, \cdots, 10\}$) with random camera orientations.

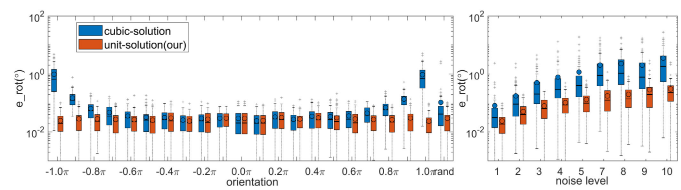

Fig. 3. Comparison of experiments using a cubic-solution and a unit-solution. Left figure: noise level was set to 1 pixel and different camera orientations were tested. Right figure: random camera orientations were tested at different noise levels.

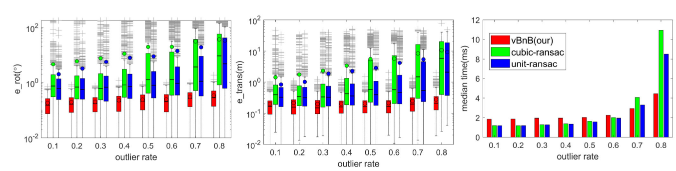

Fig. 4. Comparison of RANSAC and vBnB algorithm experiments.

The results (right subfigure in Fig. 3) show that the unit-solution is more accurate than the cubic-solution at different noise levels.

Outlier-Contaminated Inputs with a Known Vertical Direction: We then compared our proposed vBnB method with RANSAC-type methods in outlier-corrupted data with different outlier rates ($\{0.1, \cdots, 0.8\}$). The results are shown in Fig. 4. Compared with RANSAC-type methods, our proposed BnB method can obtain a globally optimal camera rotation. Besides, unit-ransac obtained slightly better accuracy than cubic-solution. This is because cubic-ransac applied the singularity-affect parameterization. Lastly, our proposed vBnB was slower than other RANSAC-type methods in low outlier rate cases, however, it was faster in high outlier rate cases, which was consistent with a conclusion that BnB-based method could be more efficient than RANSAC-type methods in high outlier rate cases [37].

Outlier-Contaminated inputs with a Biased Vertical Direction: To simulate the cases that the vertical direction was obtained inaccurately, we also tested our proposed vBnB method using synthetic data when the vertical direction was given with

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on July 27,2022 at 09:52:32 UTC from IEEE Xplore. Restrictions apply.

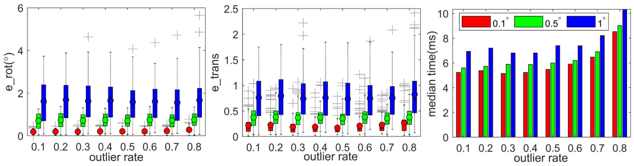

Fig. 5. Controlled experiments with different vertical direction biases.

Fig. 6. Selected images from VGG dataset.

TABLE I
EXPERIMENTS WITH REAL VGG DATA

Methods	max $e_{rot}(°)$	max $e_{trans}(m)$	avg. time (s)
Ansar + MLESAC4 + RPnL	179.12	46.90	0.719
Mirzaei + MLESAC3	178.85	46.84	0.025
RPnL + MLESAC4	177.95	32.97	0.067
P3L + RANSAC3	10.42	2.31	0.014
ASPnL + RANSAC4	2.25	1.11	0.024
LPnL_Bar_LS + AOR	178.21	32.43	0.010
LPnL_Bar_ENull + AOR	2.55	16.72	0.029
DLT-Lines + AOR	174.02	159.97	**0.008**
DLT-Plucker-Lines + AOR	179.40	122.61	0.012
DLT-Combined-Lines + AOR	178.11	2439.63	0.020
Ro_PnL (**Our**)	**1.56**	**0.84**	5.825

a bias. The outlier rate was $\{0.1, \cdots, 0.8\}$ and the vertical direction was biased by $\{0.1°, 0.5°, 1°\}$. The results are illustrated in Fig. 5. It shows that due to the biased vertical direction, the final accuracy will also be biased, however, vBnB is still robust to outliers.

C. Real-Data Experiments

In this section, we investigated the performance on the real data from VGG Multiview Dataset[4] (see Fig. 6). The VGG dataset contained image sequences, line segments, true correspondences and a ground-truth projection matrix. Outliers were randomly added to the original data, and for each scene the outlier rate was 0.2. To emphasize the global optimality, maximum rotation error, maximum translation error and average runtime were recorded for each method.

The results are listed in Table I. Our proposed algorithm could obtain the right poses ($e_{rot} < 2°$ and $e_{trans} < 1$ m) in all scenes. Conversely, other PnL algorithms could not provide all satisfactory solutions. On the other hand, to obtain robust solutions, our algorithm had longer runtime than the other existing methods.

Next, we assumed that the vertical direction was known in each scene and compared our vBnB method and RANSAC-type methods. Random outliers were also added to the clean data and outlier rate was set to 0.2. We then recorded the maximum errors and average runtime in experiments. The results are present in Table II.

TABLE II
EXPERIMENTS WITH REAL VGG DATA WITH KNOWN VERTICAL DIRECTION

Methods	max $e_{rot}(°)$	max $e_{trans}(m)$	avg. time (ms)
Cubic-ransac	57.61	43.61	3.03
Unit-ransac	11.61	3.79	**2.65**
vBnB (**Our**)	**1.51**	**0.53**	3.74

Our proposed vBnB method can obtain satisfactory solutions in all VGG data. However, RANSAC-type methods may return incorrect solutions occasionally. Moreover, the runtime of all methods in Table II were usually faster than those in Table I. This is because with the help of the prior vertical direction, the rotation domain is reduced from three to one dimension, and thus the BnB-based algorithm is much faster [33].

VII. CONCLUSION

To provide a certifiably optimal solution of PnL problem in some safety-critical applications, we propose globally optimal solutions to the camera orientation problem. The BnB algorithm is applied to search for the optimal rotation. In addition, if the vertical direction is known by other means, we first propose a novel non-minimal outlier-free PnL algorithm, which applies singular-free parameterization and thus achieves improved accuracy. Furthermore, for outlier-contaminated inputs, we propose a non-RANSAC and globally optimal algorithm to estimate camera orientation with a known vertical direction. Experiments on synthetic and real-world data all have demonstrated that our proposed methods are more robust than several existing PnL methods.

[4]https://www.robots.ox.ac.uk/~vgg/data/mview/

REFERENCES

[1] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.

[2] C. Xu, L. Zhang, L. Cheng, and R. Koch, "Pose estimation from line correspondences: A complete analysis and a series of solutions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1209–1222, Jun. 2017.

[3] T. Lemaire and S. Lacroix, "Monocular-vision based SLAM using line segments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2791–2796.

[4] N. Andreff, B. Espiau, and R. Horaud, "Visual servoing from lines," *Int. J. Robot. Res.*, vol. 21, no. 8, pp. 679–699, 2002.

[5] L. Lecrosnier, R. Boutteau, P. Vasseur, X. Savatier, and F. Fraundorfer, "Camera pose estimation based on PnL with a known vertical direction," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3852–3859, Oct. 2019.

[6] B. Přibyl, P. Zemčík, and M. Čadík, "Absolute pose estimation from line correspondences using direct linear transformation," *Comput. Vis. Image Understanding*, vol. 161, pp. 130–144, 2017.

[7] H. Yu, W. Zhen, W. Yang, and S. Scherer, "Line-based camera pose estimation in point cloud of structured environments," *CoRR*, vol. abs/1912.05013, 2019. [Online]. Available: http://arxiv.org/abs/1912.05013

[8] H. Zhou, T. Zhang, and J. Jagadeesan, "Re-weighting and 1-point RANSAC-based PnP solution to handle outliers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3022–3033, Dec. 2019.

[9] 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*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on July 27,2022 at 09:52:32 UTC from IEEE Xplore. Restrictions apply.

[10] H. M. Le, T.-J. Chin, A. Eriksson, T.-T. Do, and D. Suter, "Deterministic approximate methods for maximum consensus robust fitting," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2019.2939307.

[11] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "SE-sync: A certifiably correct algorithm for synchronization over the special Euclidean group," *The Int. J. Robot. Res.*, vol. 38, no. 2–3, pp. 95–125, 2019.

[12] V. Tzoumas, P. Antonante, and L. Carlone, "Outlier-robust spatial perception: Hardness, general-purpose algorithms, and guarantees," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 5383–5390.

[13] J. Fredriksson, V. Larsson, C. Olsson, O. Enqvist, and F. Kahl, "Efficient algorithms for robust estimation of relative translation," *Image Vis. Comput.*, vol. 52, pp. 114–124, 2016.

[14] Y. Liu, X. Li, M. Wang, A. Knoll, G. Chen, and Z. Song, "A novel method for the absolute pose problem with pairwise constraints," *Remote Sens.*, vol. 11, no. 24, 2019, Art. no. 3007.

[15] N. Horanyi and Z. Kato, "Multiview absolute pose using 3D-2D perspective line correspondences and vertical direction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2472–2480.

[16] S. Choi, T. Kim, and W. Yu, "Performance evaluation of RANSAC family," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 81.1–81.12, doi: 10.5244/C.23.81.

[17] Z. Cai, T.-J. Chin, and V. Koltun, "Consensus maximization tree search revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1637–1645.

[18] Z. Cai, T.-J. Chin, H. Le, and D. Suter, "Deterministic consensus maximization with biconvex programming," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 685–700.

[19] E. Brachmann *et al.*, "DSAC-differentiable RANSAC for camera localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6684–6692.

[20] E. Brachmann and C. Rother, "Neural-guided RANSAC: Learning where to sample model hypotheses," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 4322–4331.

[21] R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *CVGIP: Image Understanding*, vol. 60, no. 3, pp. 313–342, 1994.

[22] Y. Liu, T. S. Huang, and O. D. Faugeras, "Determination of camera location from 2-D to 3-D line and point correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 28–37, Jan. 1990.

[23] Y. Zhang, X. Li, H. Liu, and Y. Shang, "Probabilistic approach for maximum likelihood estimation of pose using lines," *IET Comput. Vis.*, vol. 10, no. 6, pp. 475–482, 2016.

[24] H. Abdellali, R. Frohlich, and Z. Kato, "A direct least-squares solution to multi-view absolute and relative pose from 2D-3D perspective line pairs," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 2119–2128.

[25] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 578–589, May 2003.

[26] P. Wang, G. Xu, Y. Cheng, and Q. Yu, "Camera pose estimation from lines: A fast, robust and general method," *Mach. Vis. Appl.*, pp. 1–12, 2019.

[27] L. Ferraz, X. Binefa, and F. Moreno-Noguer, "Very fast solution to the PnP problem with algebraic outlier rejection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 501–508.

[28] 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 *Proc. Digital Image Comput.: Techn. Appl.*, Canberra, Australia, 2018, pp. 1–8.

[29] N. Horanyi and Z. Kato, "Generalized pose estimation from line correspondences with known vertical direction," in *Proc. Int. Conf. 3D Vis.*, Qingdao, 2017, pp. 244–253.

[30] L. Lecrosnier, R. Boutteau, P. Vasseur, X. Savatier, and F. Fraundorfer, "Vision based vehicle relocalization in 3D line-feature map using perspective-n-line with a known vertical direction," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 1263–1269.

[31] F. M. Mirzaei and S. I. Roumeliotis, "Globally optimal pose estimation from line correspondences," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 5581–5588.

[32] T.-J. Chin and D. Suter, "The maximum consensus problem: Recent algorithmic advances," *Synth. Lectures Comput. Vision*, vol. 7, no. 2, pp. 1–194, 2017.

[33] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning," *Discrete Optim.*, vol. 19, pp. 79–102, 2016.

[34] A. B. Parra, T. Chin, A. Eriksson, H. Li, and D. Suter, "Fast rotation search with stereographic projections for 3D registration." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2227–2240, Nov. 2016.

[35] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A globally optimal solution to 3D ICP point-set registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.

[36] W. A. De Graaf, *Lie Algebras: Theory and Algorithms*. Amsterdam, The Netherlands: Elsevier, 2000.

[37] Á. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2868–2882, Dec. 2018.

[38] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the pnp problem: A fast, general and optimal solution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2344–2351.

[39] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, 1982. [Online]. Available: https://academic.microsoft.com/paper/1669104078

[40] J.-C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys, "Globally optimal line clustering and vanishing point estimation in manhattan world," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 638–645.

[41] H. Li, "Consensus set maximization with guaranteed global optimality for robust geometry estimation," in *Proc. IEEE 12th Int. Conf. Comput. Vision.*, 2009, pp. 1074–1080.

[42] P. H. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[43] L. Zhang, C. Xu, K.-M. Lee, and R. Koch, "Robust and efficient pose estimation from line correspondences," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 217–230.

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on July 27,2022 at 09:52:32 UTC from IEEE Xplore. Restrictions apply.