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 important 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].

To address the PnL problem, one of the most important pre-conditions 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 hypothesis-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 $t \in \mathbb{R}^3$) and orientation (i.e., rotation $R \in SO(3)$), 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.
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.

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

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

\[
\mathbf{n}_i^T R \mathbf{v}_w = 0 \tag{1}
\]

Furthermore, there is another constraint for translation \( t \).

\[
\mathbf{n}_i^T (R \mathbf{v}_w + t) = 0; \tag{2}
\]

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

\[
\min_{R} \sum_{i=1}^{N} (\mathbf{n}_i^T R \mathbf{v}_w)^2 \tag{3a}
\]

\[
\min_{t} \sum_{i=1}^{N} (\mathbf{n}_i^T (R \mathbf{v}_w + t))^2 \tag{3b}
\]
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].

\[
\max_R \sum_{i=1}^{N} \mathbb{I}\left(\|n_i^T R v_i^w\| \leq \varepsilon\right) \tag{4a}
\]

\[
\max_{R} \sum_{i=1}^{N} \mathbb{I}\left(\|n_i^T (R p_i^w + t)\| \leq \varepsilon\right) \tag{4b}
\]

where \(\varepsilon\) 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].

\section{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 \(R \in SO(3)\) corresponds to a 3-vector \(r \in \mathbb{R}^3\) whose direction and norm specify the axis and angle of rotation. Their relationship follows Rodrigues’ rotation formula:

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

where \(I\) is the 3 \(\times\) 3 identity matrix; \(\theta = ||r||\) is the angle of the rotation; \(\hat{r} = r/||r||\) 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 \(\pi\)-ball [35]. For ease of manipulation, a cube enclosing the \(\pi\)-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 \(r_0\), the lower bound of the optimum can be set as

\[
e_{L} = \sum_{i=1}^{N} \mathbb{I}\left(\|n_i^T R_0 v_i^w\| \leq \varepsilon\right) \tag{6}
\]

where \(R_0\) is the matrix form of rotation \(r_0\).

\textbf{Proof:} The function value at a specific point (i.e., \(R_0\)) within the domain must be less than the maximum value.

On the other hand, given a divided cube-shape rotation branch, whose center is \(r_0\) and whose side length is \(\phi\), the upper bound of the optimum can be set as

\[
e_{U} = \sum_{i=1}^{N} \mathbb{I}\left(\|n_i^T R_0 v_i^w\| \leq \cos(\arccos(\varepsilon) - \psi)\right) \tag{7}
\]

where \(R_0\) is the matrix form of rotation \(r_0\); \(\psi \triangleq \min\{\sqrt{3}\phi/2, \pi\}\); \([\cdot]\) is a non-negative function, which means if \(\theta < 0\), \([\theta] = 0\) and if \(\theta \geq 0\), \([\theta] = \theta\).

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

\textbf{Lemma 1:} For an arbitrary vector \(v \in \mathbb{R}^3\) and two rotations \(R_r\) and \(R_{v_0}\) in matrix form, \(r\) and \(r_0\) in angle-axis form, then

\[
\angle (R_r v, R_{v_0} v) \leq ||r - r_0|| \tag{8}
\]

where \(\angle(\cdot, \cdot)\) is the angle of two vectors. According to Lemma 1, given a divided cube-shape rotation branch, whose center is \(r_0\) and side is \(\phi\), we have

\[
\angle (R_r v_i^w, R_{v_0} v_i^w) \leq \min(||r - r_0||, \pi) \tag{9}
\]

\[
\leq \min\left\{\sqrt{3} \phi/2, \pi\right\} \triangleq \psi \tag{10}
\]

where \(r\) is an arbitrary point in the cube, whose matrix form is \(R_r\). According to triangle inequality

\[
\angle (n_i, R_r v_i^w) \leq \angle (n_i, R_{v_0} v_i^w) + \angle (R_r v_i^w, R_{v_0} v_i^w) \tag{11}
\]

\[
\leq \angle (n_i, R_{v_0} v_i^w) + \psi \tag{12}
\]

\[
\angle (n_i, R_r v_i^w) \geq \angle (n_i, R_{v_0} v_i^w) - \angle (R_r v_i^w, R_{v_0} v_i^w) \tag{13}
\]

\[
\geq \angle (n_i, R_{v_0} v_i^w) - \psi \tag{14}
\]

Hence,

\[
\angle (n_i, R_{v_0} v_i^w) - \psi \leq \angle (n_i, R_r v_i^w) \leq \angle (n_i, R_{v_0} v_i^w) + \psi \tag{15}
\]

Observe

\[
\mathbb{I}\left(\|n_i^T R_0 v_i^w\| \leq \varepsilon\right) \tag{16}
\]

\[
\Leftrightarrow \mathbb{I}\left(-\varepsilon \leq n_i^T R_0 v_i^w \leq \varepsilon\right) \tag{17}
\]

\[
\Leftrightarrow \mathbb{I}\left(-\varepsilon \leq \cos(\angle (n_i, R_{v_0} v_i^w)) \leq \varepsilon\right) \tag{18}
\]

\[
\Leftrightarrow \mathbb{I}\left(\arccos(\varepsilon) \leq \angle (n_i, R_{v_0} v_i^w) \leq \pi - \arccos(\varepsilon)\right) \tag{19}
\]

Also, we note that

\[
\arccos(\varepsilon) \leq \angle (n_i, R_{v_0} v_i^w) \leq \pi - \arccos(\varepsilon) \tag{20}
\]

\[
\Rightarrow \arccos(\varepsilon) \leq \angle (n_i, R_{v_0} v_i^w) \leq \angle (n_i, R_{v_0} v_i^w) + \psi \tag{21}
\]

\[
\Rightarrow \angle (n_i, R_{v_0} v_i^w) - \psi \leq \angle (n_i, R_{v_0} v_i^w) \leq \pi - \arccos(\varepsilon) \tag{22}
\]

\[
\Rightarrow \arccos(\varepsilon) - \psi \leq \angle (n_i, R_{v_0} v_i^w) \leq \pi - \arccos(\varepsilon) + \psi \tag{23}
\]
\[ |\arccos(\varepsilon) - \psi| \leq \angle(n_i, R_i v_0^w) \leq \pi - |\arccos(\varepsilon) - \psi| \]
\[ \Rightarrow -\cos(|\arccos(\varepsilon) - \psi|) \leq \cos \left( \angle(n_i, R_i v_0^w) \right) \leq \cos(|\arccos(\varepsilon) - \psi|) \]
\[ \Rightarrow -\cos(|\arccos(\varepsilon) - \psi|) \leq n_i^T R_i v_i^w \]
\[ \Rightarrow \leq \cos(|\arccos(\varepsilon) - \psi|) \]
\[ \Rightarrow |n_i^T R_i v_i^w| \leq \cos(|\arccos(\varepsilon) - \psi|) \]
\[ \text{where Eq. (24) follows} \]
\[ \forall i \leq N \]
\[ \leq \sum_{i=1}^{N} \varepsilon_i = \sum_{i=1}^{N} (c_i - a_i) q_i^2 + 2b_i q_i + a_i + c_i \]
\[ \Rightarrow \text{equality constrained optimization problem} [39]. \]
\[ \text{The Lagrangian formulation is} \]
\[ \frac{\max}{\min} \sum_{i=1}^{N} (a_i \cdot q_i + b_i \cdot q_i + c_i)^2, \quad \text{s.t.} \quad q_i^2 + q_i^2 = 1 \]
\[ \text{In contrast to the linear solution, it does not drop the unit-norm constraint. Additionally, in contrast to the cubic polynomial solution, it does not suffer from any degeneracy.} \]
\[ \text{Clearly, Eq. (41) is a typical equality constrained optimization problem [39]. The Lagrangian formulation is} \]
\[ f = \sum_{i=1}^{N} (a_i \cdot q_i + b_i \cdot q_i + c_i)^2 + \lambda(q_i^2 + q_i^2 - 1) \]
\[ \text{where} \ \lambda \ \text{is a Lagrange multiplier. The first-order optimality condition of the Lagrangian formulation is} \]
\[ \left\{ \begin{array}{l}
 f_q' = 2 \sum_{i=1}^{N} a_i^2 q_i + 2 \sum_{i=1}^{N} b_i q_i + 2 \sum_{i=1}^{N} c_i c_i + 2q_i = 0 \\
 f_q'' = 2 \sum_{i=1}^{N} b_i q_i + 2 \sum_{i=1}^{N} a_i q_i + 2 \sum_{i=1}^{N} b_i c_i + 2q_i = 0 \\
 f_c' = q_i^2 + q_i^2 - 1 = 0 \\
 \end{array} \right. \]
\[ \Rightarrow \left\{ \begin{array}{l}
 \tilde{a} q_i^2 + \tilde{b} q_i + \tilde{c} q_i + \tilde{d} q_i = 0 \\
 q_i^2 + q_i^2 - 1 = 0 \\
 \end{array} \right. \]
\[ \text{where} \ \tilde{a} = \sum_{i=1}^{N} a_i b_i; \ \tilde{b} = -\sum_{i=1}^{N} a_i b_i; \ \tilde{c} = \sum_{i=1}^{N} (b_i^2 - a_i^2); \ \tilde{d} = \sum_{i=1}^{N} b_i c_i. \text{The original optimal solution} (q_1, q_2) \text{can be obtained by identifying all solutions of Eq. (44).} \]
\[ \text{C. Outlier-Contaminated Cases} \]
\[ \text{In the outlier-contaminated cases, just one line correspondence is sufficient to be a minimal-subset solver in RANSAC [30]. For the} \ i-th \ \text{line correspondence, if the unit norm} \]
constraint is not ignored, then we have
\[ a_i \cdot q_1 + b_i \cdot q_2 + c_i = 0 \]
\[ q_1^2 + q_2^2 = 1 \]
\[ \Rightarrow (a_i^2 + b_i^2)q_1^2 + 2a_i c_i q_1 + c_i^2 - b_i^2 = 0 \]  \hspace{1cm} (45)

We can obtain two solutions for \( \alpha \) from Eq. (45). Similarly, two \( \alpha \) 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 certifyably optimal solution, we apply the BnB algorithm to obtain the optimal \( \alpha \) from outlier-contaminated inputs. According to Eq. (4a) and Eq. (38), the robust objective can be formulated as

\[ \max_{\alpha} \sum_{i=1}^{N} I(|a_i \cdot \cos(\alpha) + b_i \cdot \sin(\alpha) + c_i| \leq \varepsilon) \]  \hspace{1cm} (46)

Equivalently,

\[ \max_{\alpha} \sum_{i=1}^{N} I(|\sqrt{a_i^2 + b_i^2} \cdot \sin(\alpha + \varphi_i) + c_i| \leq \varepsilon) \]  \hspace{1cm} (47)

where \( \varphi_i = \arctan(2(a_i/b_i)) \).

Given a branch \([\alpha, \bar{\alpha}]\), the upper bound and lower bound can be

\[ \bar{e}_L = \sum_{i=1}^{N} I(|\sqrt{a_i^2 + b_i^2} \cdot \sin(\hat{\alpha} + \varphi_i) + c_i| \leq \varepsilon) \]  \hspace{1cm} (48)

\[ \bar{e}_U = \sum_{i=1}^{N} I(|\sqrt{a_i^2 + b_i^2} \cdot \sin(\alpha^* + \varphi_i) + c_i| \leq \varepsilon) \]  \hspace{1cm} (49)

where \( \hat{\alpha} = 0.5(\alpha + \bar{\alpha}) \) and by Interval Analysis (see [40])

\[ \alpha^* = \arg \max_{\alpha} I(|\sqrt{a_i^2 + b_i^2} \cdot \sin(\alpha + \varphi_i) + c_i| \leq \varepsilon) \]

\section{V. Translation Estimation}

After obtaining the optimal camera rotation \( R \), 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.

\section{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.

\subsection{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].
- Mirzaeei+MLESAC3: Mirzaeei PnL algorithm [31] is nested inside MLESAC.
- RPnL+MLESAC4: RPnL algorithm [43] is nested inside MLESAC.
- P3L+RANSAC4: A P3L algorithm [2] is plugged into RANSAC.
- ASPnL+RANSAC4: ASPnL algorithm [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 al. in [2].
- DLT-Pücker-Lines+AOR: It applies Pücker coordinates [6].
- DLT-Combined-Lines+AOR: It is a combination of DLT-Lines and DLT-Pücker-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 \( \varepsilon \) 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: \( e_{\text{trans}} = ||\mathbf{g}_t - \mathbf{t}^*|| \) where \( \mathbf{g}_t \) is ground truth; \( \mathbf{t}^* \) is the estimated translation. The rotation/orientation error is defined as the angle between ground truth \( \mathbf{R}_g \) and estimated rotation \( \mathbf{R}^* \): \( e_{\text{rot}} = \arccos(0.5 \star \text{trace}((\mathbf{R}_g^{-1} \mathbf{R}^*) - 1)) \). To compare the efficiency, the median runtime for 500 trials is recorded for each setting.

\(^1\)\( \arctan(2(\cdot, \cdot)) \) 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/
B. Synthetic Data Experiments

$N = 200$ line segments were created randomly using $2N$ random endpoints, which were distributed in a cube $[100 \text{ m} \times 100 \text{ m} \times 100 \text{ m}]^3$. A virtual pinhole camera was randomly placed in the scene facing towards the line segments. The camera is simulated using a $640 \times 480$ 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 $e_{\text{rot}} < 5^\circ$ and $e_{\text{trans}} < 2 \text{ m}$. 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 ($\alpha = \{-\pi, \ldots, \pi\}$) 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 ($\{1, \ldots, 10\}$) with random camera orientations.

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 ($\{0.1, \ldots, 0.8\}$). 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...
a bias. The outlier rate was \([0.1, \cdots, 0.8]\) and the vertical direction was biased by \([0.1^\circ, 0.5^\circ, 1^\circ]\). 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 \((\varepsilon_{\text{rot}} < 2^\circ \text{ and } \varepsilon_{\text{trans}} < 1 \text{ m})\) 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.

![Selected images from VGG dataset.](image)

**Fig. 6.** Selected images from VGG dataset.

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 the 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.

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].

![Controlled experiments with different vertical direction biases.](image)

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

<table>
<thead>
<tr>
<th>Methods</th>
<th>(\max \varepsilon_{\text{rot}}(\text{(^\circ)}))</th>
<th>(\max \varepsilon_{\text{trans}}(\text{m}))</th>
<th>avg. time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansar + MLESAC4 + RPNl.</td>
<td>179.12</td>
<td>46.90</td>
<td>0.719</td>
</tr>
<tr>
<td>Mirzae + MLESAC3</td>
<td>178.85</td>
<td>46.84</td>
<td>0.025</td>
</tr>
<tr>
<td>RPNl. + MLESAC4</td>
<td>177.95</td>
<td>32.97</td>
<td>0.067</td>
</tr>
<tr>
<td>P3L. + RANSAC3</td>
<td>10.42</td>
<td>2.31</td>
<td>0.014</td>
</tr>
<tr>
<td>A5Phnl. + RANSAC4</td>
<td>2.25</td>
<td>1.11</td>
<td>0.024</td>
</tr>
<tr>
<td>LPhnl_2x_2 + AOR</td>
<td>178.21</td>
<td>32.43</td>
<td>0.010</td>
</tr>
<tr>
<td>LPhnl_2x_2Wall + AOR</td>
<td>2.55</td>
<td>16.72</td>
<td>0.029</td>
</tr>
<tr>
<td>DLT-Lines + AOR</td>
<td>174.02</td>
<td>159.97</td>
<td>0.008</td>
</tr>
<tr>
<td>DLT-Pickler-Lines + AOR</td>
<td>179.40</td>
<td>122.61</td>
<td>0.012</td>
</tr>
<tr>
<td>DLT-Combined-Lines + AOR</td>
<td>178.11</td>
<td>2439.63</td>
<td>0.020</td>
</tr>
<tr>
<td>Rv_PnF. (Our)</td>
<td>1.56</td>
<td>0.84</td>
<td>5.825</td>
</tr>
</tbody>
</table>

**Table II**

**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.

**REFERENCES**


