2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

A Fast and Accurate Solution for Pose Estimation from 3D Correspondences

Lipu Zhou, Shengze Wang, and Michael Kaess

Abstract— **Estimating pose from given 3D correspondences, including point-to-point, point-to-line and point-to-plane correspondences, is a fundamental task in computer vision with many applications. We present a fast and accurate solution for the least-squares problem of this task. Previous works mainly focus on studying the way to find the global minimizer of the least-squares problem. However, existing works that show the ability to achieve the global minimizer are still unsuitable for real-time applications. Furthermore, as one of contributions of this paper, we prove that there exist ambiguous configurations for any number of lines and planes. These configurations have several solutions in theory, which makes the correct solution may come from a local minimizer when the data are with noise. Previous works based on convex optimization which is unable to find local minimizers do not work in the ambiguous configuration. Our algorithm is efficient and able to reveal local minimizers. We employ the Cayley-Gibbs-Rodriguez (CGR) parameterization of the rotation to derive a general rational cost for the three cases of 3D correspondences. The main contribution of this paper is to solve the first-order optimality conditions of the least-squares problem, which are of a complicated rational form. The central idea of our algorithm is to introduce some intermediate unknowns to simplify the problem. Extensive experimental results show that our algorithm is more stable than previous algorithms when the number N of correspondences is small. Besides, when N is large, our algorithm achieves the same accuracy as the state-of-the-art algorithm [1], but our algorithm is about 7 times faster than [1] in real applications.**

I. Introduction

Estimating the pose from 3D correspondences, *i.e.* point-to-point, point-to-line and point-to-plane correspondences, is known as the 3D registration problem in the literature [2], [3], [4], [5], [1], [6]. It is one of the fundamental problems in computer vision with a wide range of applications, such as simultaneous localization and mapping (SLAM) [7], [8], [9], [10], [11], extrinsic calibration [12], [13], [14], [15], [16] and iterative closes point (ICP) framework [17]. Besides, some camera pose estimation problems, such as the perspective-n-point (PnP) problem [18], [19], [20] and the perspective-n-line (PnL) problem [21], [22], can be transformed to a 3D registration problem [6]. However, the research on this problem is not as thorough as other pose estimation problems.

Previous works mainly focus on solving the least-squares problem. Although large progress has been made, achieving the optimal solution and the real-time performance is still a challenge. Some algorithms [4], [1] are capable of finding the global minimum, however, their running time makes them not suitable for real-time applications. Recently, Wientapper [6] provided an efficient algorithm, but their algorithm can not achieve the global optimality in theory, and has the risk of no solution. Furthermore, previous works generally assume that the correct solution is the global minimizer. However, we find that the correct solution may come from a local minimizer in some configurations. The contributions of this paper are as follows:

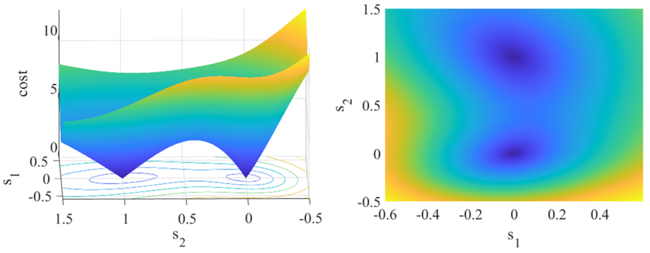

Fig. 1. An ambiguous configuration for four point-to-line correspondences which has 2 solutions, described in lemma 1. We plot the side view (left) and the top view (right) of the cost (12) for the unknowns s_1 and s_2 defined in (5). To the best of our knowledge, this is the first work revealing the existence of the ambiguity for the 3D registration problem.

First, we prove that there exist ambiguous configurations for any number of plane and line correspondences, which leads to multiple solutions in theory. When a configuration approximates ambiguity, the correct solution of the problem may come from a local minimizer. Therefore, previous works [2], [4], [1] that are only able to compute the global minimizer may fail in these cases. Revealing local minimizers is essential for an algorithm to handle all the configurations of the 3D registration problem.

Second, we present an efficient and accurate solution for the 3D registration problem. We use the Cayley-Gibbs-Rodriguez (CGR) parametrization to represent the rotation, which generates a rational cost function. We derive its first-order optimality conditions. They form a high order polynomial system, and are hard to solve. Four intermediate unknowns are introduced to relax the original problem, which results in much simpler first-order optimality conditions. Gröbner basis method [23] is applied to solve this equation system. Then we refine the solution by the Newton-Raphson method.

We evaluate our algorithm with synthetic and real data. Extensive experimental results show that our algorithm significantly outperforms previous works [2], [4], [1], [6]when the number of correspondences is small. Besides, experimental results verify that our algorithm can converge to the global minimizer with the same accuracy as the previous works with guaranteed globally optimality [4], [1], however, our

This work was partially supported by the Office of Naval Research under grant N00014-16-1-2103.

Lipu Zhou, Shengze Wang and Michael Kaess are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA zhoulipu@outlook.com,{shengzew,kaess}@andrew.cmu.edu

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

algorithm is much faster.

II. Related Works

This paper focuses on the pose estimation from point-to-point, point-to-line and point-to-plane correspondences correspondences. Pose estimation from point-to-point correspondences has been solved in early works [24], [25], [26]. There exist closed-form solutions for this problem. However, estimating pose from point-to-line and point-to-plane correspondences is more complicated. These 3D correspondences actually yield similar distance function, which is employed by previous works to construct a general cost function [2], [3], [4], [5], [1], [6]. The difference of various cost functions lies in the parameterization of the rotation matrix. The raw rotation matrix is adopted in [3], [1], and the quaternion [25] is used in [2], [4], [5], [6]. Previous works mainly focus on finding the global minimizer of the resulting cost function. In [4], they proposed a provably optimal algorithm. They employed convex underestimators with branch-and-bound methods to iteratively compute the global minimum of the cost function. Although this algorithm can guarantee global optimality, it is very time-consuming. Olsson *et al.* [3] reduced the computational complexity by applying the Lagrangian dual relaxation to approximate the cost function. Their experimental results showed that a single convex semidefinite program can well approximate the original problem. Recently, an improved result is obtained by a strengthened Lagrangian dual relaxation [1]. Although without theoretical guarantees, their experimental results show that this algorithm can achieve the global minimum. Although these algorithms [3], [1] have made progress in efforts to reduce the computational time, they are still not suitable to demanding real-time applications. Recently, Wientapper [6] provided an efficient algorithm based on Gröbner basis polynomial solver. They introduced the first order derivatives of the norm-one constraint of the quaternion into the first order optimality conditions of the cost function as [20], rather than apply the Lagrangian formulation for the norm-one constraint of the quaternion. This results in an efficient solution, however, their result is not optimal in theory. Furthermore, as the number of equations is greater than the number of unknowns, the equation system may have no solution. They introduced 4 pre-rotations to solve the problem 4 times to handle this problem. The pre-rotation is to make the no-solution problem happen less likely, however, it can not fully solve this problem in theory.

3D registration without correspondence is an related but more complicated problem. Several works [27], [28], [29], [30] presented globally optimal solution for the 3D point cloud registration without correspondence. The ICP framework [17] gives a general way to find correspondences and the pose at the same time. Although this framework was originally designed for the point, other 3D models can also be introduced into this framework [31], [32], [33]. The ICP framework iteratively finds the nearest elements as correspondences then calculates the pose until it converges. Efficiently and accurately estimating the pose from current 3D correspondences is critical for the ICP framework.

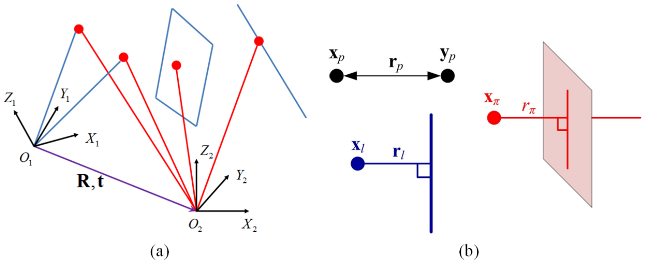

Fig. 2. (a) Pose estimation for mixed point-to-point, point-to-line and point-to-plane correspondences. (b) A schematic of the point-to-point, point-to-line and point-to-plane distance.

III. Problem Formulation

In this paper, we use italic, boldfaced lowercase and boldfaced uppercase letters to represent scalars, vectors and matrices, respectively. This paper focuses on the problem of pose estimation from point-to-plane, point-to-line and point-to-point correspondences as demonstrated in Fig. 2 (a). Fig. 2 (b) illustrates the distance of the three correspondences. We first consider the point-to-plane correspondence. 2 (b). Suppose we have a point $\mathbf{x}_l$ in one coordinate system and the corresponding plane in another coordinate system represented by the plane's norm-one normal $\mathbf{n}$ and a point $\mathbf{y}^\pi$ lying on it. Then the scalar residual between a point and a plane can be written as

$$r_\pi = \mathbf{n}^T \left(\mathbf{R}\mathbf{x}_\pi + \mathbf{t} - \mathbf{y}_\pi\right) \tag{1}$$

For the point-to-line correspondence, we can calculate the 3-dimensional residual vector between a point $\mathbf{x}_l$ and the corresponding line represented by the norm-one direction $\mathbf{d}$ and a point $\mathbf{y}_l$ on it as

$$\mathbf{r}_l = \left(\mathbf{I}_3 - \mathbf{d}\mathbf{d}^T\right)\left(\mathbf{R}\mathbf{x}_l + \mathbf{t} - \mathbf{y}_l\right) \tag{2}$$

where $\mathbf{I}_3$ is the identify matrix. Lastly, a point-to-point correspondence yields a 3-dimensional residual vector

$$\mathbf{r}_p = \mathbf{R}\mathbf{x}_p + \mathbf{t} - \mathbf{y}_p \tag{3}$$

$\mathbf{R}$ and $\mathbf{t}$ have 6 DoF. We adopt the concept, *i.e.* effective number of correspondences, defined in [1]. Let us denote the number of effective correspondences as N. One point-to-plane, point-to-line and point-to-point correspondence provide 1, 2, and 3 constraints, respectively. For n_π point-to-plane, n_l point-to-line, and n_p point-to-point correspondences, we have $N = n_\pi + 2 \times n_l + 3 \times n_p$. This paper seeks to estimate the pose for $N > 6$.

IV. Least Squares Solution

Using the notation in (1), (2) and (3), we define our cost function as follows

$$\begin{aligned} &C_{\pi lp}\left(\mathbf{R},\mathbf{t}\right) = \sum_{i=1}^{n_\pi} r_{\pi_i}^2 + \sum_{i=1}^{n_l} \mathbf{r}_{l_i}^T \mathbf{r}_{l_i} + \sum_{i=1}^{n_p} \mathbf{r}_{p_i}^T \mathbf{r}_{p_i}, \\ &s.t.\ \mathbf{R}\mathbf{R}^T = \mathbf{I}_3, \det\left(\mathbf{R}\right) = 1, \end{aligned} \tag{4}$$

where $\det\left(\mathbf{R}\right)$ represents the determinant of $\mathbf{R}$.

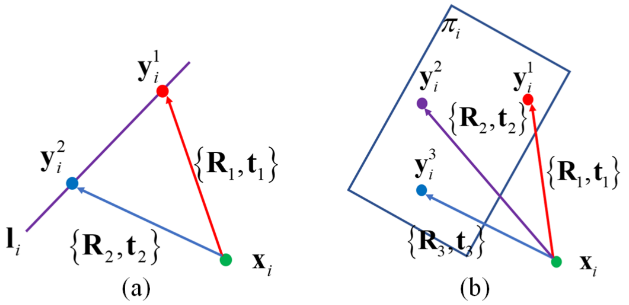
$\mathbf{y}_i^1$
$\mathbf{y}_i^2$
$\mathbf{l}_i$
$\{\mathbf{R}_2,\mathbf{t}_2\}$
$\{\mathbf{R}_1,\mathbf{t}_1\}$
$\mathbf{x}_i$
(a)
π_i
$\mathbf{y}_i^2$
$\mathbf{y}_i^1$
$\{\mathbf{R}_2,\mathbf{t}_2\}$
$\mathbf{y}_i^3$
$\{\mathbf{R}_1,\mathbf{t}_1\}$
$\{\mathbf{R}_3,\mathbf{t}_3\}$
$\mathbf{x}_i$
(b)

Fig. 3. A schematic of a point-to-line ambiguous configuration (a) and a point-to-plane ambiguous configuration (b) described in **Lemma 1** and **2**, respectively.

A. Ambiguous Configurations

In previous works, the global minimizer is treated as the optimal solution. It is well known that there is an unique solution for at least three point-to-point correspondences, except for the degenerate configuration, *i.e.* all the points are collinear. However, for plane and line, there exist configurations which have several solutions. We call them as ambiguous configurations. The ambiguous configuration differs from the degenerate configuration. The ambiguous configuration has several solutions, but the degenerate configuration has infinite solutions. For ambiguous cases, the global minimizer can only find one solution, which may miss the correct solution. Formally, we have the following lemma for point-to-line correspondences.

Lemma 1: For any n_l points and any $\{\mathbf{R}_1,\mathbf{t}_1\}$ and $\{\mathbf{R}_2,\mathbf{t}_2\}$, there exist n_l lines to make $\{\mathbf{R}_1,\mathbf{t}_1\}$ and $\{\mathbf{R}_2,\mathbf{t}_2\}$ the exact solutions for the n_l point-to-line correspondences.

Proof: Define $\mathbf{x}_i$ the ith point of an arbitrary point set with n_l points. For any $\{\mathbf{R}_1,\mathbf{t}_1\}$ and $\{\mathbf{R}_2,\mathbf{t}_2\}$, we can have $\mathbf{y}_i^1 = \mathbf{R}_1\mathbf{x}_i + \mathbf{t}_1$ and $\mathbf{y}_i^2 = \mathbf{R}_2\mathbf{x}_i + \mathbf{t}_2$. Then we can use $\mathbf{y}_i^1$ and $\mathbf{y}_i^2$ to define a line $\mathbf{l}_i = [\mathbf{d}_i;\mathbf{y}_i]$ with direction $\mathbf{d}_i = \frac{\mathbf{y}_i^1-\mathbf{y}_i^2}{\|\mathbf{y}_i^1-\mathbf{y}_i^2\|_2}$ and a point $\mathbf{y}_i = \mathbf{y}_i^1$ on it, as demonstrated in Fig. 3 (a). According to how we construct $\mathbf{l}_i$, we know that $\mathbf{l}_i$ passes through $\mathbf{y}_i^1$ and $\mathbf{y}_i^2$. Therefore $\{\mathbf{R}_1,\mathbf{t}_1\}$ and $\{\mathbf{R}_2,\mathbf{t}_2\}$ are two solutions for the n_l point-to-line $\{\mathbf{x}_i \leftrightarrow \mathbf{l}_i\}_{i=1}^{n_l}$. ■

Similarly, we have lemma 2 for point-to-plane correspondences.

Lemma 2: For any n_π points and any $\{\mathbf{R}_1,\mathbf{t}_1\}$, $\{\mathbf{R}_2,\mathbf{t}_2\}$ and $\{\mathbf{R}_3,\mathbf{t}_3\}$, there exist n_π planes to make $\{\mathbf{R}_1,\mathbf{t}_1\}$, $\{\mathbf{R}_2,\mathbf{t}_2\}$ and $\{\mathbf{R}_3,\mathbf{t}_3\}$ are exact solutions for the n_π point-to-plane correspondences.

Proof: Define $\mathbf{x}_i$ the ith point of an arbitrary point set with n_π points. For any $\{\mathbf{R}_1,\mathbf{t}_1\}$, $\{\mathbf{R}_2,\mathbf{t}_2\}$ and $\{\mathbf{R}_3,\mathbf{t}_3\}$, we can have $\mathbf{y}_i^1 = \mathbf{R}_1\mathbf{x}_i + \mathbf{t}_1$, $\mathbf{y}_i^2 = \mathbf{R}_2\mathbf{x}_i + \mathbf{t}_2$ and $\mathbf{y}_i^3 = \mathbf{R}_3\mathbf{x}_i + \mathbf{t}_3$. Then we can find a plane π_i passing through $\mathbf{y}_i^1$, $\mathbf{y}_i^2$ and $\mathbf{y}_i^3$, as demonstrated in Fig. 3 (b). According to how we construct π_i, we know that $\{\mathbf{R}_1,\mathbf{t}_1\}$, $\{\mathbf{R}_2,\mathbf{t}_2\}$ and $\{\mathbf{R}_3,\mathbf{t}_3\}$ are three solutions for the n_π point-to-plane correspondences $\{\mathbf{x}_i \leftrightarrow \pi_i\}_{i=1}^{n_\pi}$. ■

Finally, we have the following theorem for n_l point-to-line and n_π point-to-plane correspondences.

Theorem 1: For any n_l points on lines and n_π points on planes and any $\{\mathbf{R}_1,\mathbf{t}_1\}$ and $\{\mathbf{R}_2,\mathbf{t}_2\}$, there exist n_l lines and n_π planes to make $\{\mathbf{R}_1,\mathbf{t}_1\}$ and $\{\mathbf{R}_2,\mathbf{t}_2\}$ are exact solutions for the n_l point-to-line and n_π point-to-plane correspondences.

Proof: We first consider the n_l points on lines. Define $\mathbf{x}_i^l$ is the ith point. According to Lemma 1, we can find n_l lines to make $\{\mathbf{R}_1,\mathbf{t}_1\}$ and $\{\mathbf{R}_2,\mathbf{t}_2\}$ are exact solutions for the n_l point-to-line correspondences $\{\mathbf{x}_i^l \leftrightarrow \mathbf{l}_i\}_{i=1}^{n_l}$.

Then we consider the n_π points on planes. For the jth point $\mathbf{x}_j^\pi$ within them, we can define ${}^1\mathbf{y}_j^\pi = \mathbf{R}_1\mathbf{x}_j^\pi + \mathbf{t}_1$ and ${}^2\mathbf{y}_j^\pi = \mathbf{R}_2\mathbf{x}_j^\pi + \mathbf{t}_2$. Let us denote π_j as a plane passing through the line defined by ${}^1\mathbf{y}_j^\pi$ and ${}^2\mathbf{y}_j^\pi$. According to how we construct π_j, we know that $\{\mathbf{R}_1,\mathbf{t}_1\}$ and $\{\mathbf{R}_2,\mathbf{t}_2\}$ are two solutions for the n_π point-to-plane correspondences $\{\mathbf{x}_j^\pi \leftrightarrow \pi_j\}_{j=1}^{n_\pi}$.

Therefore, $\{\mathbf{R}_1,\mathbf{t}_1\}$ and $\{\mathbf{R}_2,\mathbf{t}_2\}$ are the two solutions for the n_l point-to-line correspondences $\{\mathbf{x}_i^l \leftrightarrow \mathbf{l}_i\}_{i=1}^{n_l}$ and n_π point-to-plane correspondences $\{\mathbf{x}_i^\pi \leftrightarrow \pi_i\}_{i=1}^{n_\pi}$. ■

When measurements approximate an ambiguous configuration, a prior is required to identify the correct solution. The prior is generally available in real applications. For instance, we generally have a rough estimation of the pose between two sensors in the extrinsic calibration problem [12], [13], [14], [15], [16]. For the pose estimation in SLAM [7], [8], [9], [10], [11], current pose should be consistent with the previous motion trajectory.

Previous works [2], [3], [4], [1] using convex approximation can only find the global minimizer. However, the global minimizer may not be the correct solution for a problem when measurements approximate an ambiguous configuration. Therefore, we can not simply omit local minimizers.

B. Rotation Parameterization

Solving for the rotation matrix is the crux for pose estimation. Previous works [2], [4], [1], [6] adopt non-minimal representations for $\mathbf{R}$, which results in additional quadratic constraints in the minimization problem. This paper adopts the CGR parametrization [18], [34] which gives a minimal representation for the rotation matrix, removing the quadratic constraints in (4) . The CGR parametrization expresses a rotation matrix as

$$\mathbf{R} = \frac{\bar{\mathbf{R}}}{1+\mathbf{s}^T\mathbf{s}},\ \bar{\mathbf{R}} = \left(\left(1-\mathbf{s}^T\mathbf{s}\right)\mathbf{I}_3 + 2[\mathbf{s}]_\times + 2\mathbf{s}\mathbf{s}^T\right) \quad (5)$$

where $\mathbf{s} = [s_1; s_2; s_3]$ is a 3-dimensional vector, and $[\mathbf{s}]_\times$ is the skew-symmetric matrix of $\mathbf{s}$.

C. Rational Form of the Residual

We will show that the residuals from point-to-point, point-to-line and point-to-plane correspondences have a general form

$$r_g = \mathbf{a}^T\mathbf{R}\mathbf{b} + \mathbf{a}^T\mathbf{t} + c \quad (6)$$

where $\mathbf{a}$ and $\mathbf{b}$ are two 3-dimensional vectors and c is a scalar. We first consider the point-to-plane residual r_π in (1). It is obvious that $\mathbf{a} = \mathbf{n}$, $\mathbf{b} = \mathbf{y}_\pi$ and $c = -\mathbf{n}^T\mathbf{y}_\pi$. Let us then consider the point-to-line residual. We define $\mathbf{r}_l = \left[r_l^1, r_l^2, r_l^3\right]^T$ and $\mathbf{A}_l = \left(\mathbf{I}_3 - \mathbf{d}\mathbf{d}^T\right) = \left[\mathbf{a}_l^1; \mathbf{a}_l^2; \mathbf{a}_l^3\right]$, where $\mathbf{a}_l^1, \mathbf{a}_l^2$ and $\mathbf{a}_l^3$ are the three rows of $\mathbf{A}_l$. Using this notation, equation (2) can be written as

$$r_l^i = \mathbf{a}_l^i \left(\mathbf{R}\mathbf{x}_l + \mathbf{t} - \mathbf{y}_l\right),\ i = 1, 2, 3 \tag{7}$$

It is easy to find that $\mathbf{a} = \left(\mathbf{a}_l^i\right)^T,\ \mathbf{b} = \mathbf{x}_l$ and $c = -\mathbf{a}_l^i \mathbf{y}_l$. Lastly, we consider the point-to-point correspondence. Similar to the point-to-line correspondence, we define $\mathbf{r}_p = \left[r_p^1, r_p^2, r_p^3\right]^T$ and $\mathbf{I}_3 = \left[\mathbf{e}_l^1; \mathbf{e}_l^2; \mathbf{e}_l^3\right]$. Substituting this notation into (3), it is easy to find

$$r_p^i = \mathbf{e}_p^i \left(\mathbf{R}\mathbf{x}_p + \mathbf{t} - \mathbf{y}_p\right),\ i = 1, 2, 3 \tag{8}$$

The similarity between (8) and (7) is obvious. Therefore, r_p^i also has the general form (6).

Let $\mathbf{t} = [t_1; t_2; t_3]$. Substituting (5) into the general residual r_g in (6) and adding the tree terms together yields

$$r_g = \frac{\mathbf{a}^T \bar{\mathbf{R}} \mathbf{b}}{1 + \mathbf{s}^T \mathbf{s}} + \mathbf{a}^T \mathbf{t} + c = \frac{\mathbf{k}^T \mathbf{v}}{1 + \mathbf{s}^T \mathbf{s}} \tag{9}$$

where $\mathbf{k}^T \mathbf{v}$ is a third order polynomial with terms $\mathbf{v} = [s_1^2 t_1, s_1^2 t_2, s_1^2 t_3, s_1^2, s_1 s_2, s_1 s_3, s_1, s_2^2 t_1, s_2^2 t_2, s_2^2 t_3, s_2^2, s_2 s_3, s_2, s_3^2 t_1, s_3^2 t_2, s_3^2 t_3, s_3^2, s_3, t_1, t_2, t_3, 1]^T$.

D. First-Order Optimality Conditions

Now we consider the least-squares problem (4). Using (9), the squared residual can be represented as

$$r_g^2 = \frac{q\left(\mathbf{s}, \mathbf{t}\right)}{\left(1 + \mathbf{s}^T \mathbf{s}\right)^2} \tag{10}$$

where $q\left(\mathbf{s}, \mathbf{t}\right) = \mathbf{v}^T \mathbf{k} \mathbf{k}^T \mathbf{v}$ is a 6th order polynomial in $\mathbf{s}$ and $\mathbf{t}$. For the point-to-line distance $\mathbf{r}_{l_i}^T \mathbf{r}_{l_i}$ in (4), we have

$$\mathbf{r}_{l_i}^T \mathbf{r}_{l_i} = \left(r_{l_i}^1\right)^2 + \left(r_{l_i}^2\right)^2 + \left(r_{l_i}^3\right)^2 \tag{11}$$

As mentioned above, $r_{l_i}^1$, $r_{l_i}^2$ and $r_{l_i}^3$ all have the same form as r_g. Therefore, the point-to-line distance $\mathbf{r}_{l_i}^T \mathbf{r}_{l_i}$ will have the same form as (10). Similarly, $\mathbf{r}_p^T \mathbf{r}_p$ will also yield the same form as (10). After the summation of squared residuals in (4), we know that $C_{\pi l p}$ would have the same rational form as (10). We write it as

$$C_{\pi l p}\left(\mathbf{s}, \mathbf{t}\right) = \frac{\bar{C}_{\pi l p}\left(\mathbf{s}, \mathbf{t}\right)}{\left(1 + \mathbf{s}^T \mathbf{s}\right)^2} \tag{12}$$

where $\bar{C}_{\pi l p}$ is a 6th order polynomial function in $\mathbf{s}$ and $\mathbf{t}$. To find the critical points of (12), we first calculate its first order optimality conditions as follows:

$$\begin{aligned} g_{s_i} &= \tfrac{\partial C_{\pi l p}}{\partial s_i} = \tfrac{1}{(1+\mathbf{s}^T\mathbf{s})^3} \left(\left(1 + \mathbf{s}^T \mathbf{s}\right) \tfrac{\partial \bar{C}_{\pi l p}}{\partial s_i} - 4 s_i \bar{C}_{\pi l p}\right) = 0 \\ g_{t_i} &= \tfrac{\partial C_{\pi l p}}{\partial t_i} = \tfrac{1}{(1+\mathbf{s}^T\mathbf{s})^2} \tfrac{\partial \bar{C}_{\pi l p}}{\partial t_i} = 0,\ i = 1, 2, 3 \end{aligned} \tag{13}$$

Canceling the denominator of (13) yields

$$\begin{aligned} \bar{g}_{s_i} &= \left(1 + \mathbf{s}^T \mathbf{s}\right) \tfrac{\partial \bar{C}_{\pi l p}}{\partial s_i} - 4 s_i \bar{C}_{\pi l p} = 0 \\ \bar{g}_{t_i} &= \tfrac{\partial \bar{C}_{\pi l p}}{\partial t_i} = 0,\ i = 1, 2, 3 \end{aligned} \tag{14}$$

$\frac{\partial \bar{C}_{\pi l p}}{\partial s_i}$ and $\frac{\partial \bar{C}_{\pi l p}}{\partial t_i}$ are of degree 5. Therefore, $\bar{g}_{s_i}$ and $\bar{g}_{t_i}$ are of degree 7 and 5, respectively. Although the Gröbner basis method gives a general way to solve the polynomial system, it is computationally demanding and numerically unstable to apply it to a high order polynomial system, as the experimental results in Fig. 4. The Newton-Raphson method provides an alternative way to find the roots of (14). Denote an equation system as $\bar{G}\left(\mathbf{x}\right)$. For the kth iteration, the Newton-Raphson method updates the solution as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - J_{\bar{G}}^{-1}\left(\mathbf{x}_{k-1}\right) \bar{G}\left(\mathbf{x}_{k-1}\right) \tag{15}$$

This iterative method requires an initial solution $\mathbf{s}_0$ and $\mathbf{t}_0$. In the next section, we will describe how to calculate an accurate $\mathbf{s}_0$ and $\mathbf{t}_0$.

E. Initial Estimation from Relaxation

The difficulty of solving (14) lies in the denominator of CGR parameterization (5). To solve this problem, we introduce the following intermediate variable

$$\rho = \frac{1}{\sqrt{1 + s_1^2 + s_2^2 + s_3^2}}. \tag{16}$$

Substituting (16) into (9) will transform (9) from a rational function to a polynomial function. Furthermore, we define

$$\alpha = \rho s_1,\ \beta = \rho s_2,\ \gamma = \rho s_3. \tag{17}$$

Using (16) and (17), the residual (9) can be expressed as

$$r_g = \mathbf{a}_g^T \mathbf{u} + \mathbf{b}_g{}^T \mathbf{t} \tag{18}$$

where $\mathbf{u} = \left[\alpha^2, \alpha\beta, \alpha\gamma, \alpha\rho, \beta^2, \beta\gamma, \beta\rho, \gamma^2, \gamma\rho, \rho^2, 1\right]^T$. This formulation turns the rational function (9) into a polynomial function (18) that is easier to handle. Furthermore, $\mathbf{t}$ in (18) is decoupled from $\mathbf{R}$ as in [6], [1], and (18) is linear in $\mathbf{t}$.

Now we consider the least-squares problem (4) for the new unknowns. We stack the residuals from n_π point-to-plane, n_l point-to-line, and n_p point-to-point correspondences to get

$$\mathbf{e}_g = \mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{t} \tag{19}$$

where $\mathbf{A}$ is a $(n_\pi + 3n_l + 3n_p) \times 11$ matrix and $\mathbf{B}$ is a $(n_\pi + 3n_l + 3n_p) \times 3$ matrix. Then (4) can be written as

$$C_{\pi l p}\left(\rho, \alpha, \beta, \gamma, \mathbf{t}\right) = \mathbf{e}_g^T \mathbf{e}_g \tag{20}$$

As (19) is linear for $\mathbf{t}$, then $\mathbf{t}$ has a closed-form solution as

$$\mathbf{t} = -\left(\mathbf{B}^T \mathbf{B}\right)^{-1} \mathbf{B}^T \mathbf{A} \mathbf{u} \tag{21}$$

Substituting (21) into (20), we derive a cost function only involving $\rho, \alpha, \beta, \gamma$ as

$$C_{\pi l p}\left(\rho, \alpha, \beta, \gamma\right) = \mathbf{u}^T \mathbf{C}^T \mathbf{C} \mathbf{u} = \mathbf{u}^T \mathbf{Q} \mathbf{u} \tag{22}$$

where $\mathbf{C} = \mathbf{A} - \mathbf{B}\left(\mathbf{B}^T \mathbf{B}\right)^{-1} \mathbf{B}^T \mathbf{A}$. The elements in the vector $\mathbf{u}$ are second order monomials except for the constant term. Thus (22) is a fourth order polynomial function. We compute the first order optimality conditions to get all the stationary points. This first-order condition contains four third order polynomials for ρ, α, β and γ as

$$\begin{aligned} g_\rho &= \frac{\partial C_{\pi l p}}{\partial \rho} = 0, \quad g_\alpha = \frac{\partial C_{\pi l p}}{\partial \alpha} = 0, \\ g_\beta &= \frac{\partial C_{\pi l p}}{\partial \beta} = 0, \quad g_\gamma = \frac{\partial C_{\pi l p}}{\partial \gamma} = 0. \end{aligned} \tag{23}$$

The solution of (23) includes the global and local minimizers of $C_{\pi lp}$. According to Bézout's theorem [23], there exist at most $3^4 = 81$ solutions. We find that the polynomial system in (23) only contains third and first degree monomials. This equation system stratifies the 2-fold symmetry [35], [36], [19]. That is to say if any nontrivial $\xi = [\rho, \alpha, \beta, \gamma]^T$ is a solution, $-\xi$ is also a solution. Thus there are at most 40 independent solutions. We adopt the algorithm introduced in [37] to generate the polynomial solver which can utilize this property to yield an efficient solution.

After solving (23), we are then able to compute $\mathbf{t}$ from (21). The CGR parameters s_1, s_2 and s_3 can be recovered from the definitions in (16) and (17). The above formulation treats $\rho, \alpha, \beta, \gamma$ as independent unknowns. However, they are related as they are functions of s_1, s_2 and s_3. To recover the minimizer of (12), we refine the solution using the Newton-Raphson iteration method in (15). We summarize our least-squares solution in Algorithm 1.

```
Algorithm 1 Least-squares solution
Input: $n_\pi$ point-to-plane, $n_l$ point-to-line, and $n_p$ point-to-point correspondences
Output: $\mathbf{R}$ and $\mathbf{t}$
  1. Compute the coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ in (19).
  2. Compute $\mathbf{Q}$ in (22).
  3. Compute the first order optimality conditions (23)
  4. Solve the equation system (23) for $\rho, \alpha, \beta, \gamma$.
  5. Recover $s_1, s_2, s_3$ using (17).
  6. Compute $\mathbf{t}$ from (21).
  7. Refine the solution by Newton-Raphson iteration (15).
```

V. Experiments

In this section, we compare our algorithm with the state-of-the-art algorithms, including Briales's algorithm [1], BnB [4], Olsson's algorithm [3] and Wientapper's algorithm [6].

A. Experiments with Synthetic Data

Our synthetic data are generated as [1]. Specifically, each geometric element is determined by randomly sampling a point within a sphere of radius $10m$. For lines and planes, a random unit direction and normal are generated. We uniformly sample the Euler angles φ, ϑ, ψ of the rotation matrix ($\varphi, \psi \in [0°, 360°]$ and $\vartheta \in [0°, 180°]$). The translation elements are uniformly distributed within $[-10m, 10m]$. We use $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$ to represent the estimated rotation and translation and use $\mathbf{R}_{gt}$ and $\mathbf{t}_{gt}$ to represent the ground truth. The rotation error is evaluated by the angle of the axis-angle representation of $\mathbf{R}_{gt}^{-1}\mathbf{R}$, and the translation error is evaluated by $\left\|\mathbf{t}_{gt} - \hat{\mathbf{t}}\right\|_2 / \|\mathbf{t}_{gt}\|$. We consider the effective number of the correspondences as [1]. For n_π point-to-plane, n_l point-to-line, and n_p point-to-point correspondences, the number of correspondences for is calculated as $N = n_\pi + 2n_l + 3n_p$. Given an N, we randomly generate a combination of point, line and point whose effective number is N.

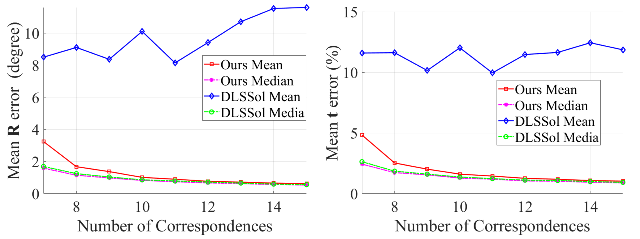

Fig. 4. Compare our least-squares solution with the direct least-squares solution (DLSSol) which directly solves the polynomial system (14).

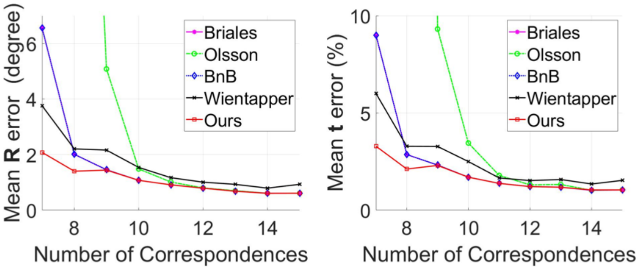

Fig. 5. Rotation and translation errors for increasing number of correspondences. The noise level is fixed to 0.05m.

Effect of our intermediate unknowns As it is generally difficult to get a stable solution for a high order polynomial system [23], we introduce 4 intermediate unknowns to simplify the least-squares problem. To verify their benefit, we evaluate the performance of the direct least-squares solution (DLSSol) for the polynomial system (14). We employed the algorithm in [37] to generate the solver. We ran 2000 trials for each $N \in [7, 15]$. Fig. 4 shows the results. Directly solving (14) can recover the static points of cost function (12) in theory. However, the large mean errors of DLSSol verify that this polynomial solver is very unstable. Furthermore, the computational time of DLSSol and our algorithm are 20.0 ms and 2.95 ms, respectively. Our solution is about 7 times faster than DLSSol. The above results verify that our intermediate unknowns can increase the numerical stability as well as significantly reduce the computational time.

Least-squares solution We conduct experiments to evaluate the performances of different algorithms under varying number of correspondences, increasing level of noise and computational time. The results of all experiments are from 100 independent trials as [1].

The first experiment considers a fixed noise level and an

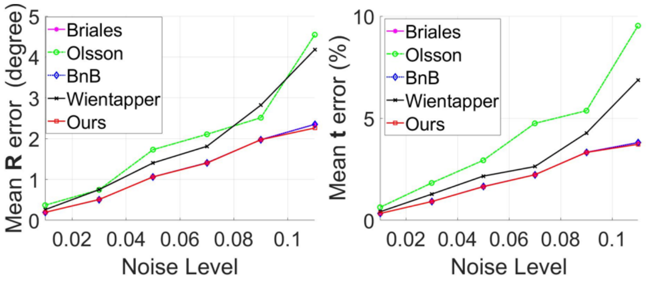

Fig. 6. Rotation and translation errors for increasing noise level. The number of effective correspondences is set to 10.

TABLE I

EXPERIMENTAL RESULTS ON THE 03, 04, AND 07 SEQUENCES OF THE KITTI DATASET [38].

Method	KITTI 03			KITTI 04			KITTI 07		
	$\mathbf{t}$ Err (%)	$\mathbf{R}$ Err (°)	Time (s)	$\mathbf{t}$ Err (%)	$\mathbf{R}$ Err (°)	Time (s)	$\mathbf{t}$ Err (%)	$\mathbf{R}$ Err (°)	Time (s)
Olsson [3]	2.03e-1	**5.46e-1**	2.22	5.75e-1	5.76e-1	1.88	**3.60e-1**	**2.53e-1**	2.08
Wientapper [6]	2.00e-1	5.72e-1	0.58	4.32e-1	3.49e-1	0.55	3.61e-1	2.58e-1	0.56
Briales [1]	1.96e-1	5.61e-1	2.16	**4.31e-1**	**3.37e-1**	1.88	**3.60e-1**	**2.53e-1**	2.03
Ours	**1.95e-1**	5.61e-1	**0.33**	**4.31e-1**	**3.37e-1**	**0.30**	**3.60e-1**	**2.53e-1**	**0.30**

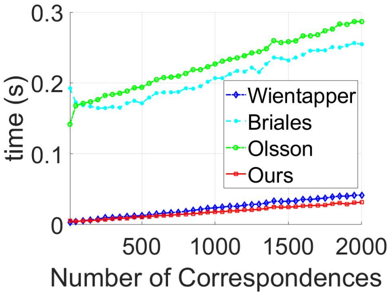

Fig. 7. Computational time of different algorithms.

TABLE II

AVERAGE NUMBERS OF POINT-TO-POINT(P-TO-P), POINT-TO-PINE(P-TO-L), AND POINT-TO-PLANE(P-TO-PL) CORRESPONDENCES PER FRAME FOR 03, 04 AND 07 KITTI SEQUENCES.

Sequences	Average Correspondences Per Frame		
	#P-to-P	#P-to-L	#P-to-PL
KITTI 03	63	17	21117
KITTI 04	37	30	19994
KITTI 07	111	40	20126

increasing number of correspondences. Let us denote the standard deviation of a zero mean Gaussian noise distribution as δ. We set $\delta = 0.05m$. N varies from 7 to 15. Fig. 5 shows the result. The results of BnB [4] and Briales's algorithm [1] overlap. This is consistent to the results in [1]. The compared algorithms are all unstable for a small N. Our algorithm significantly outperforms them when N is small. When N is large, our algorithm achieves the same accuracy as [4], [1]. Wientapper's method [6] can not find the optimal solution even N is large.

In the second experiment, δ is from $0.01m$ to $0.11m$, stepping by 0.02m. The results are illustrated in Fig. 6. BnB, Briales's algorithm and our algorithm present better results than other algorithms. Our algorithm gives a slightly better result than BnB and Briales's algorithm when $\delta = 0.11m$.

In the last experiment, we compare the computational time of different algorithms. We vary the effective number of correspondences N from 10 to 2000. For every N, we run each algorithm 100 times to calculate the average running time. Fig. 7 provides the result. We did not run the BnB algorithm [4], because it is too slow for a large N. It is clear that our algorithm is the fastest one among the compared algorithms.

B. Experiments with Real Data

We generated our real-world dataset from the KITTI dataset [38]. The characteristics of the dataset is in Table II. There are more than 20,000 correspondences in each frame, with the majority being the point-to-plane correspondence.

For each frame, 2D feature points are detected by the ORB feature detector [39]. We project LiDAR points into the image plane, and select the LiDAR points around an ORB feature. Then, we fit a plane to these LiDAR points. Finally, we compute the 3D coordinates of an ORB feature by calculating the intersection of the back-projection ray of the ORB feature and the plane. We use the ORB descriptor to match with the 2D feature points in the previous frame, then we obtain the 3D point-to-point correspondences.

Next, we generate the point-to-line correspondence. For each frame, 2D lines are detected by the Line Segment Detector (LSD) [40] and described by the Line Band Descriptor (LBD) [41]. A 2D line is represented by two endpoints. The two 3D endpoints of a line are calculated as it is done for the 2D feature point described above. We generate 3D line correspondences by matching their 2D LBD features. Given a line-to-line correspondence, two point-to-line correspondences are generated for the 2 endpoints.

Finally, we get the point-to-plane correspondence. For each frame, planes are extracted from the LiDAR point cloud by the region growing algorithm [42]. Then we find the nearest plane point in the previous frame to generate the point-to-plane correspondence.

Given the initial correspondences as described above, we run at most 3 ICP iterations to handle the outlier. In each iteration, we eliminate the correspondences with distance larger than 5cm as outliers. We evaluate the performance of different algorithms on sequences 03, 04, and 07 of the KITTI dataset. We did not test BnB [4], as it is extremely slow on this large dataset. Our algorithm achieves the same or slightly better result as the state-of-the-art algorithm [1] while being about 7 times faster, as shown in Table I.

VI. CONCLUSIONS

This paper presents an efficient and accurate solution for the 3D registration problem. We proof that there exists ambiguous configurations for any number of point-to-plane and point-to-line correspondences. This requires an algorithm should have the ability to reveal local minimizers. We use the CGR parameterization to represent the rotation, removing the constraints on the rotation. However, this results in a rational form cost which is hard to solve. We introduce 4 intermediate variables to simplify the first order optimality conditions. We evaluate our algorithm through synthetic and real data. The experimental results show that our algorithm is more stable for a small N, and is as accurate as the state-of-the-art algorithm [1] when N is large, but is much faster.

References

[1] J. Briales and J. Gonzalez-Jimenez, "Convex global 3d registration with lagrangian duality," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] C. Olsson, F. Kahl, and M. Oskarsson, "The registration problem revisited: Optimal solutions from points, lines and planes," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 1206–1213.

[3] C. Olsson and A. Eriksson, "Solving quadratically constrained geometrical problems using lagrangian duality," in *2008 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–5.

[4] C. Olsson, F. Kahl, and M. Oskarsson, "Branch-and-bound methods for euclidean registration problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 783–794, 2009.

[5] F. Wientapper and A. Kuijper, "Unifying algebraic solvers for scaled euclidean registration from point, line and plane constraints," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 52–66.

[6] F. Wientapper, M. Schmitt, M. Fraissinet-Tachet, and A. Kuijper, "A universal, closed-form approach for absolute pose problems," *Computer Vision and Image Understanding*, vol. 173, pp. 57–75, 2018.

[7] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2174–2181.

[8] ——, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.

[9] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011, pp. 127–136.

[10] P. F. Proença and Y. Gao, "Probabilistic rgb-d odometry based on points, lines and planes under depth uncertainty," *Robotics and Autonomous Systems*, vol. 104, pp. 25–39, 2018.

[11] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane slam for hand-held 3d sensors," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5182–5189.

[12] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[13] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09*, 2005.

[14] O. Naroditsky, A. Patterson, and K. Daniilidis, "Automatic alignment of a camera with a line scan lidar system," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3429–3434.

[15] R. Gomez-Ojeda, J. Briales, E. Fernandez-Moral, and J. Gonzalez-Jimenez, "Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3611–3616.

[16] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5562–5569.

[17] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–607.

[18] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (dls) method for pnp," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 383–390.

[19] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the pnp problem: A fast, general and optimal solution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2344–2351.

[20] L. Kneip, H. Li, and Y. Seo, "Upnp: An optimal o (n) solution to the absolute pose problem with universal applicability," in *European Conference on Computer Vision*. Springer, 2014, pp. 127–142.

[21] 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*, vol. 39, no. 6, pp. 1209–1222, 2017.

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

[23] D. A. Cox, J. Little, and D. O'shea, *Using algebraic geometry*. Springer Science & Business Media, 2006, vol. 185.

[24] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[25] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Josa a*, vol. 4, no. 4, pp. 629–642, 1987.

[26] B. K. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *JOSA A*, vol. 5, no. 7, pp. 1127–1135, 1988.

[27] H. Li and R. Hartley, "The 3d-3d registration problem revisited," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[28] C. Papazov and D. Burschka, "Stochastic global optimization for robust point set registration," *Computer Vision and Image Understanding*, vol. 115, no. 12, pp. 1598–1609, 2011.

[29] J. Yang, H. Li, and Y. Jia, "Go-icp: Solving 3d registration efficiently and globally optimally," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1457–1464.

[30] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision*. Springer, 2016, pp. 766–782.

[31] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. IEEE, 2015, pp. 742–749.

[32] A. Censi, "An icp variant using a point-to-line metric," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 19–25.

[33] J. Serafin and G. Grisetti, "Nicp: Dense normal based point cloud registration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 742–749.

[34] F. M. Mirzaei and S. I. Roumeliotis, "Optimal estimation of vanishing points in a manhattan world," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2454–2461.

[35] V. Larsson and K. Åström, "Uncovering symmetries in polynomial systems," in *European Conference on Computer Vision*. Springer, 2016, pp. 252–267.

[36] E. Ask, Y. Kuang, and K. Åström, "Exploiting p-fold symmetries for faster polynomial equation solving," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 3232–3235.

[37] V. Larsson, K. Astrom, and M. Oskarsson, "Efficient solvers for minimal problems by syzygy-based reduction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 820–829.

[38] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[39] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf," 2011.

[40] G. Randall, J. Jakubowicz, R. G. von Gioi, and J. Morel, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 722–732, 12 2008. [Online]. Available: doi.ieeecomputersociety.org/10.1109/TPAMI.2008.300

[41] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[42] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, "Fast plane detection and polygonalization in noisy 3d range images," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 3378–3383.