Minimal Solutions for Pose Estimation of a Multi-Camera System

Gim Hee Lee, Bo Li, Marc Pollefeys, and Friedrich Fraundorfer

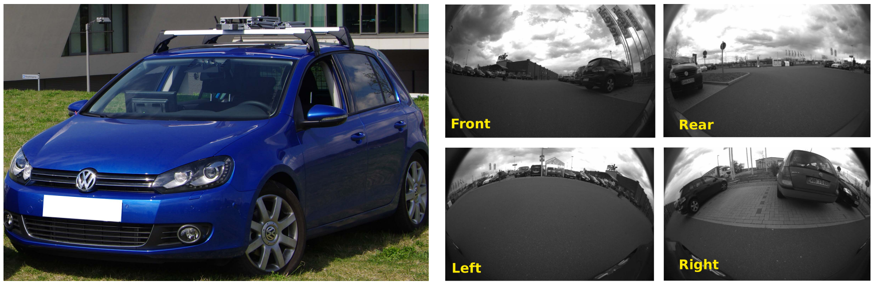
Abstract In this paper, we propose a novel formulation to solve the pose estimation problem of a calibrated multi-camera system. The non-central rays that pass through the 3D world points and multi-camera system are elegantly represented as Plücker lines. This allows us to solve for the depth of the points along the Plücker lines with a minimal set of 3-point correspondences. We show that the minimal solution for the depth of the points along the Plücker lines is an 8 degree polynomial that gives up to 8 real solutions. The coordinates of the 3D world points in the multi-camera frame are computed from the known depths. Consequently, the pose of the multi-camera system, i.e. the rigid transformation between the world and multi-camera frames can be obtained from absolute orientation. We also derive a closed-form minimal solution for the absolute orientation. This removes the need for the computationally expensive Singular Value Decompositions (SVD) during the evaluations of the possible solutions for the depths. We identify the correct solution and do robust estimation with RANSAC. Finally, the solution is further refined by including all the inlier correspondences in a non-linear refinement step. We verify our approach by showing comparisons with other existing approaches and results from large-scale real-world datasets.

1 Introduction

The pose estimation problem of a multi-camera system refers to the problem of determining the rigid transformation between the world frame and multi-camera frame, given a set of 3D points defined in the world frame and its corresponding 2D image points. In contrast with a single camera that has a single center of projection,

Gim Hee Lee, Bo Li and Marc Pollefeys
Department of Computer Science, ETH Zürich, Universitätstrasse 6 CH-8092 Zürich. e-mail: {glee@student, libo@student, marc.pollefeys@inf}.ethz.ch

Friedrich Fraundorfer
Faculty of Civil Engineering and Surveying, Technische Universität München, Arcisstrasse 21 80333 München. e-mail: friedrich.fraundorfer@tum.de



(a) (b)

Fig. 1 (a) Our robotic car platform with a multi-camera system made up of 4 separate fish-eye cameras looking front, rear, left and right (cameras are embedded in the car logos and side mirrors). (b) Sample images from the 4 cameras.

the multi-camera system is an imaging sensor where light rays passing through the 3D world points and camera are non-central, i.e. the light rays do not meet at a single center of projection. An advantage of the multi-camera system is that it provides the flexibility to be set in a configuration which gives a maximum coverage of the environment. The solution to the pose estimation problem of a multi-camera system has important applications in robotics such as finding the initial camera pose estimates in structure-from-motion (SfM) / visual Simultaneous Localization and Mapping (SLAM), geometric verification and place recognition for loop-closures, and visual localization of a robot with respect to a given map that contains visual descriptors. Figure 1 shows our robotic car platform and the images from the multi-camera system mounted on it.

The fact that the light rays from a multi-camera system do not meet at a single center of projection means that all the classical approaches [6, 17, 13] for solving the perspective pose problem cannot be used. An alternative approach has to be proposed to handle the non-central nature of the multi-camera system. In addition, it is important that the proposed approach is a minimal solution and requires minimal correspondences that makes it efficient to be used within robust estimators such as RANSAC [5] (see Section 5 for more detail).

In this paper, we proposed a novel formulation to solve the pose estimation problem of a multi-camera system. In particular, we adopt the representation of non-central light rays from a multi-camera system with the Plücker line coordinates from existing works [16, 12, 10, 11] for relative motion estimation of the multi-camera system. We show that this allows us do a two-step approach for solving the pose estimation problem - (a) solve for the unknown depth of the points along the Plücker lines and (b) compute the multi-camera pose from the known depths with absolute orientation [6, 8]. We show that with a minimal number of 3-point correspondences, it leads to an 8 degree polynomial minimal solution that yields up to a maximum of 8 real solutions for the unknown depths. Each of these possible solutions of the depth is used to compute the coordinates of the 3D world points in the multi-camera frame. The known 3D points in the multi-camera frame are used to compute the pose of the multi-camera system using absolute orientation.

The standard approach for solving the absolute orientation requires an expensive step of SVD and it is inefficient to perform the SVD multiple times to evaluate all the possible solutions of the depths. We circumvent this problem by deriving an efficient minimal solution for the absolute orientation, which allows us to compute the rigid transformation between the world and multi-camera frames from 3-point correspondences in closed-form without the need for SVD. Once we have obtained all the possible solutions for the rigid transformation, we compute the depths of all the other 3D world points. This allows us to choose the correct solution within a robust estimator such as RANSAC. Finally, the solution is further refined by including all the inlier correspondences in a non-linear refinement step that minimizes the reprojection errors (see Section 6 for more detail). We verify our approach by showing comparisons with other existing approaches and results from large-scale real-world datasets.

2 Related Works

The method proposed by Chen *et al*. [2] is most related to our method. In this work, they proposed a 3-point minimal solution and N-point solution to the multi-camera pose estimation problem. Similar to our method, their proposed solution is also a two-step approach. First, the coordinates of the 3D points in the multi-camera frame are determined. The 3D points in the multi-camera frame are determined by solving three distance parameters defined on the rays that passes through the 3D points. Next, the rigid transformation between the 3D points in the world and multi-camera frames is solved by absolute orientation. The formulation in the first step resulted in two 8 degree polynomials where a total of up to 16 real solutions are computed by root finding. In comparison, our method resulted in only one 8 degree polynomial that gives up to 8 real solutions, which has the advantage of less computational time needed to identify the correct solution. Another drawback of [2] is that the representations of the rays used to define the distance parameters breaks down when the three rays are respectively lying on parallel planes and in the case of linear pushbroom cameras [7] (see Section 4.3 for more detail). As a result, an alternative representation has to be made. In contrast, our representation of the rays as the Plücker lines is holistic and does not require any alternative formulation in any case. In addition, we also derive an efficient closed-form minimal solution for absolute orientation.

In [14], Nister proposed a formulation that directly solves for the rotation and translation parameters. His formulation gives an 8 degree polynomial minimal solution. This method is of special interest as the coefficients for the equation system can be computed with a low number of computations making it a fast method. He also proposed the use of Sturm sequencing for root finding and stated that the execution times is in the order of microseconds. The method is evaluated with simulations and compared to the single camera case. Similar to Nister's method, our method also ends up with an 8 degree polynomial minimal solution, which can also be solved with the Sturm sequencing to achieve the same execution time. Despite the computational efficiency, as also noted in [9], the derivation of Nister's method is not intuitive and requires laborious geometry and algebraic reasonings.

Kneip *et al*. presented that most recent work on pose estimation using a multi-camera system in [9]. In this work, the authors presented a 3-point minimal solution and N-point solution. They first solved for the rotations and point depths with a Gröbner Basis [3] solver followed by solving for the translation. They showed simulation experiments, comparisons to single camera perspective pose methods and a real-world visual odometry experiment using a two-camera setup. The exact process of solving the pose estimation problem with the Gröbner Basis approach is a black-box process which is not described in detail in [9]. Hence, Kneip's method cannot be reproduced easily. In comparison, our method is based on several algebraic equations which are intuitive and easy to implement. They mentioned that the generated solution from the Gröbner Basis solver has a length of 8000 lines of code and the execution time in the order of milliseconds. This makes it slower than Chen's, Nister's and our methods which solve an 8 degree polynomial that can be done in the order of microseconds as noted in Nister's paper [14].

In contrast to the minimal solvers for the pose estimation problem of the multi-camera system, there also exist linear [4] and iterative N-point [18, 19] solutions. The linear solution needs 6 or more point correspondences and thus less efficient in RANSAC [5] compared to our minimal solution which requires only 3 point correspondences. Since the iterative N-point solutions involves computationally expensive iterations, they are usually used to refine the pose estimation after all the inlier point correspondences have been found by RANSAC coupled with a minimal solution.

We adopt the Plücker lines representation for a multi-camera system from existing works on motion estimation [16, 12, 10, 11]. However, it is important to note that we adopt the Plücker lines representation to solve the multi-camera pose estimation problem, which is a completely different problem from the multi-camera motion estimation problem in [16, 12, 10, 11]. The objective of multi-camera motion estimation is to compute the relative transformation between two multi-camera frames given the 2D-2D image point correspondences, while the multi-camera pose estimation problem ask for the rigid transformation between a given world frame and the multi-camera frame given the 2D image point to 3D world point correspondences. To the best of our knowledge, no other work has adopted the Plücker lines representation to solve the multi-camera pose estimation problem.

3 Problem Definition

Figure 2 shows an illustration of the pose estimation problem of the multi-camera system. It is made up of multiple cameras denoted by (C_1, C_2, C_3) that are rigidly fixed onto a single body. Note that we show only 3 cameras in Figure 2 but our proposed method works for any multi-camera system that has any number of cameras. Our method also works even if there was only 1 single camera (see perspective case in Section 4.3). We denote the reference frame of the multi-camera system and the

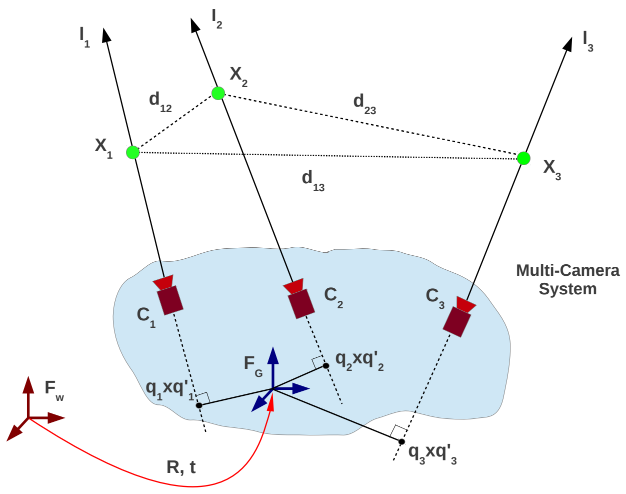


Fig. 2 Illustration of the pose estimation problem for a multi-camera system.

world frame as F_G and F_W. The intrinsics and extrinsics of the respective cameras are assumed to be known from calibration and are denoted by K_i and $T_{C_i} = [R_{C_i}\ t_{C_i};\ 0\ 1]$ with respect to the multi-camera frame F_G, where $i = 1,2,3$. The pose estimation problem of a multi-camera system is defined as follows:

Definition 1. Given a set of three 3D points defined in F_W denoted by (X_1, X_2, X_3) that are seen by arbitrary cameras on the multi-camera system and their corresponding 2D image coordinates denoted by (x_1, x_2, x_3), find the rigid transformation R and t that brings the multi-camera frame F_G into the world frame F_W.

4 Multi-Camera Pose Estimation

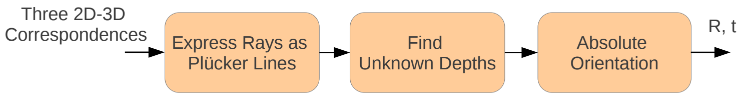


Fig. 3 Our formulation for the pose estimation of the multi-camera system.

Figure 3 shows an illustration of our formulation for pose estimation of the multi-camera system. We first express the rays that join the respective three 2D-3D correspondences as Plücker line coordinates with respect to the multi-camera frame F_G (see Section 4.1 for more detail). Next, we solve for the unknown depths associated with each of the Plücker line using our minimal solution that leads to an 8 degree polynomial giving up to 8 real solutions (see Section 4.2 for more detail). Lastly, we compute the coordinates of the 3D points in the multi-camera frame F_G with the known depths and solve for the rigid transformation R and t between the world and multi-camera frames using our efficient minimal solution of absolute orientation in closed-form (see Section 4.4 for more detail).

4.1 Plücker Line Representation

We saw in Section 1 that the main problem with a multi-camera system is the absence of a single projection center for the camera. Following [16], we remove the need for a single projection center by representing the rays that pass through the 3D world points and the multi-camera system as Plücker line coordinates expressed in the multi-camera frame F_G. The Plücker line is a 6-vector $l_i = [q_i^T \; q_i'^T]^T$ where $i = 1,2,3$ as shown in Figure 2. $q_i = R_{C_i}\hat{x}_i$ is the unit direction of the ray expressed in the multi-camera frame F_G where $\hat{x}_i = K_i^{-1}x_i$ is the normalized image coordinate of the point x_i. The closest point from the Plücker line to F_G is given by $q_i \times q_i'$ as shown in Figure 2 and it is the point that forms a perpendicular intersection on the Plücker line from the multi-camera frame F_G. q_i' is defined as the cross product $q_i' = t_{Ci} \times q_i$. Any point X_i^G that is expressed in the multi-camera frame F_G is given by

$$X_i^G = q_i \times q_i' + \lambda_i q_i \tag{1}$$

where λ_i is the depth of the point X_i^G along the Plücker line, i.e. the signed distance from $q_i \times q_i'$ to X_i^G. Note that λ always has to be positive for the 3D point to appear in front of the camera.

4.2 Minimal Solution for Depths

The distances d_{ij} where $(i,j) \in \{(1,2),(1,3),(2,3)\}$ between the 3D points X_i in the world frame F_W shown in Figure 2 have to be the same as the distances between the 3D points X_i^G in the multi-camera frame F_G , i.e.

$$||X_i - X_j||^2 = ||X_i^G - X_j^G||^2 \tag{2}$$

where $(i,j) \in \{(1,2),(1,3),(2,3)\}$. By making use of the preservation of the 3D point distances given by Equation 2 and the Plücker line equation from Equation 1, we get three constraints

$$||X_i - X_j||^2 = ||(q_i \times q_i' + \lambda_i q_i) - (q_j \times q_j' + \lambda_j q_j)||^2 \tag{3}$$

where $(i,j) \in \{(1,2),(1,3),(2,3)\}$ with three unknown depths λ_1, λ_2 and λ_3 from the Plücker lines. Expanding and rearranging the unknowns in Equation 3, we get

$$k_{11}\lambda_1^2 + (k_{12}\lambda_2 + k_{13})\lambda_1 + (k_{14}\lambda_2^2 + k_{15}\lambda_2 + k_{16}) = 0 \tag{4a}$$

$$k_{21}\lambda_1^2 + (k_{22}\lambda_3 + k_{23})\lambda_1 + (k_{24}\lambda_3^2 + k_{25}\lambda_3 + k_{26}) = 0 \tag{4b}$$

$$k_{31}\lambda_2^2 + (k_{32}\lambda_3 + k_{33})\lambda_2 + (k_{34}\lambda_3^2 + k_{35}\lambda_3 + k_{36}) = 0 \tag{4c}$$

where k are the coefficients made up from the known Plücker line coordinates q_i and q'_i, and 3D world points X_i. We drop the full expressions of the coefficients for brevity. Using the Sylvester Resultant [3] to eliminate λ_1 from Equations 4a and 4b, we get a polynomial $f(\lambda_2,\lambda_3)=0$ which is a function of only λ_2 and λ_3. We do another Sylvester Resultant on $f(\lambda_2,\lambda_3)=0$ and Equation 4c to eliminate λ_2, we get an univariate 8 degree polynomial dependent on only λ_3.

$$A\lambda_3^8+B\lambda_3^7+C\lambda_3^6+D\lambda_3^5+E\lambda_3^4+F\lambda_3^3+G\lambda_3^2+H\lambda_3+I=0 \tag{5}$$

where A,B,C,D,E,F,G,H and I are coefficients made up from k from Equation 4. The roots of Equation 5 can be obtained from the eigen-values of the Companion matrix [3] made up of the coefficients. A maximum of up to 8 real roots can be obtained for λ_3. As suggested in [14], a more efficient way to solve for the roots of the 8 degree polynomial is by using the Sturm sequences.

λ_2 can be found by back-substituting λ_3 in Equation 4c. After completing the square on Equation 4c and making λ_2 the subject, we get

$$\lambda_2=\frac{1}{2a}(-b\pm\sqrt{b^2-4ac}) \tag{6}$$

where $a=k_{31},\ b=k_{32}\lambda_3+k_{33},\ c=k_{34}\lambda_3^2+k_{35}\lambda_3+k_{36}$. Similarly, λ_1 can be found by back-substituting λ_2 into Equation 4a which takes similar form as Equation 6 after completing the square and making λ_1 the subject. A total of up to 32 (i.e. $8\times 2\times 2$) solution triplets of λ_1, λ_2 and λ_3 can be obtained. A solution triplet is discarded if any one of the λs is an imaginary or negative value. A further step to disambiguate the solutions is by doing a redundancy check on λ_1 using Equation 4b. The solution pairs of λ_2 and λ_3 should produce consistent λ_1 from both Equations 4a and 4b. Any solution pair of λ_2 and λ_3 which produces λ_1 with discrepancy from Equations 4a and 4b is discarded. In our simulations, we observed that these disambiguation checks are capable of reducing the maximum number of solutions to two for most of the time. All the other existing 2D-3D point correspondences are used to identify the correct solution within RANSAC, i.e. the correct solution yields the highest number of inliers in RANSAC.

4.3 Special Cases

In this section, we look at five special cases for the multi-camera pose estimation problem mentioned in [2]. In particular, we compare the similarities and differences between the existing methods and our method under these five different cases.

Case 1: Partially Parallel. Two out of the three light rays are parallel in this case as illustrated in Figure 4. This means that two of the unit directions must be equal, i.e. $q_1=q_2\neq q_3$. From Figure 4, we can see that $\lambda_2=\lambda_1+c_{12}$, where c_{12} is a known value from the Plücker line coordinates and distance between the 3D points (X_1, X_2). Applying the Sylvester Resultant for variable elimination together with Equations 4b and 4c, we get a 4 degree polynomial minimal solution that can be

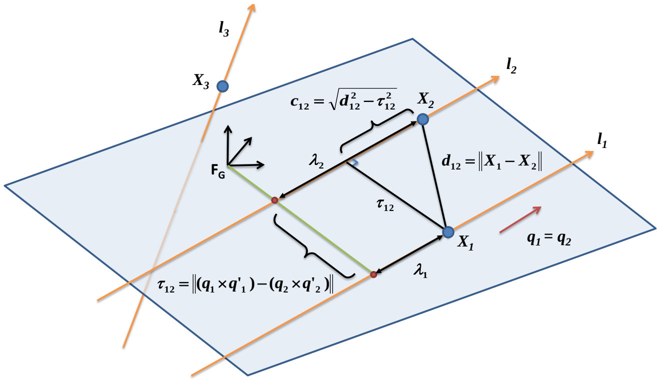


Fig. 4 Illustration of the partially parallel case.

solved in closed-form. Similar 4 degree polynomial minimal solution was obtained for Chen's and Nister's methods.

Case 2: Perspective. The three light rays pass through a common center of projection in the perspective case, i.e. all the 2D-3D correspondences are from one single camera in the multi-camera system. Let us choose the camera reference frame to be the center of projection. This implies that the camera extrinsics become $t_{C_1} = t_{C_2} = t_{C_3} = 0$ and $R_{C_1} = R_{C_2} = R_{C_3} = I$. Substituting these values into Equation 3 and applying the Sylvester Resultant for variable elimination, we get a 4 degree polynomial minimal solution that can be solved in closed-form. This result is similar to Chen's and Nister's method, and the P3P solution for a perspective camera [6]. Note that a 4 degree polynomial is obtained even if the reference frame was not chosen as the center of projection of the camera.

Case 3: Parallel Plane. This is the case where the three light rays lie on three different planes that are parallel to each other. It is important to note that these light rays however do not have the same unit direction, i.e. $q_1 \neq q_2 \neq q_3$ from the Plücker lines. It can be observed that the constraints from our method in Equation 3 does not break down. In contrast, the representations of the rays used by Chen *et al*. [2] to define the distance parameters cannot be used in the case where all the three rays respectively lie on parallel planes. As a result, an alternative representation has to be made.

Case 4: Linear Pushbroom. There is only one camera in the case of linear pushbroom [7]. Here, the camera moves through a straight line of motion with a known speed and takes images at regular intervals. Hence, the transformations between any three camera locations (similar to the extrinsics of a multi-camera system) are known and the rays that observed unique 3D world points from these locations lie on parallel planes. This implies that the linear pushbroom case is the same as the parallel plane case where our method does not break down. In comparison, an alternative representation has to be made for Chen's method.

Case 5: Orthographic. For orthographic projection, all the light rays are parallel. Hence, all the unit directions of the Plücker lines are equal, i.e. $q_1 = q_2 = q_3$. Similar to the partially parallel case, we have the following 3 constraints $\lambda_2 =$

$\lambda_1 + c_{12}$, $\lambda_3 = \lambda_1 + c_{13}$ and $\lambda_3 = \lambda_2 + c_{23}$, where infinite solutions exist for λ_1, λ_2 and λ_3. Intuitively, we can move the multi-camera system anywhere along the direction of the parallel light rays and the constraints are still fulfilled, hence infinite solutions. This degeneracy is independent of the formulation and holds for all works [2, 14, 9, 4, 18, 19] on pose estimation for the multi-camera system.

4.4 Minimal solution for Absolute Orientation

Absolute Orientation can be solved using the methods from [8, 6]. However, these methods require a computationally inefficient step of SVD which becomes an overhead when it is used numerous times within RANSAC to compute all the hypothesis solutions. We present a minimal solution which allows us to compute the absolute orientation in closed-form without the need for SVD. The proposed method computes the transformation R and t to align the two point sets P and Q consisting of three correspondence 3D points as shown in Equation 7.

$$P_i = RQ_i + t, \quad i = 1,2,3 \tag{7}$$

First, two local frames F_M and F_N are defined on the point sets P and Q respectively. The origins of the local frames are defined on the first points, i.e. P_1 and Q_1. We can now write the transformed points in the newly defined local frames F_M and F_N as $M_i = P_i - P_1$ and $N_i = Q_i - Q_1$. Next, we define the x-axis of each local frame to pass through the second point respectively, i.e. M_2 and N_2. The x-axis of F_M and F_N can be aligned by applying the following transformations

$$M_2 = \begin{bmatrix} M_{2x} \\ M_{2y} \\ M_{2z} \end{bmatrix} = R_M \begin{bmatrix} ||M_2|| \\ 0 \\ 0 \end{bmatrix}, N_2 = \begin{bmatrix} N_{2x} \\ N_{2y} \\ N_{2z} \end{bmatrix} = R_N \begin{bmatrix} ||N_2|| \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

where R_M and R_N are unknown rotation matrices that align the two x-axis. Here, we only describe the steps to solve for R_M since R_N can be computed in an analogous fashion. Since the alignment of the x-axis only involves two rotations around the y- and z-axis, R_M can be written as

$$R_M = R_{Mz} R_{My} = \begin{bmatrix} ce & -f & de \\ cf & e & df \\ -d & 0 & c \end{bmatrix} \tag{9}$$

where c and d are sine and cosine of the rotation angle around the y-axis, and e and f are sine and cosine of the rotation angle around the z-axis. Putting Equation 9 into Equation 8, we get the following three constraints

$$||M_2||ce - M_{2x} = 0 \quad (10a) \quad ||M_2||cf - M_{2y} = 0 \quad (10b) \quad -M_{2z} - ||M_2||d = 0 \quad (10c)$$

where d can be calculated directly from Equation 10c and c can then be computed with the Pythagoras identity. e and f can be solved by substituting c into Equations 10a and 10b. The full expressions for solving a, b, c and d are given as follows

$$d = \frac{-M_{2z}}{\|M_2\|},\ c = \sqrt{1-d^2},\ e = \frac{M_{2x}}{\|M_2\|c},\ f = \frac{M_{2y}}{\|M_2\|c} \tag{11}$$

Finally, we apply R_M and R_N to align the x-axis of both point sets. The new sets of transformed points are given by

$$U_i = R_M^T M_i,\ V_i = R_N^T N_i, \quad i = 1{,}2{,}3 \tag{12}$$

The last step is to find the remaining rotation R_V around the x-axis which would complete the alignment of the two local frames F_M and F_N. This gives the constraint in Equation 13 which can be expanded into three independent constraints in Equations 14a-14c.

$$U_3 = R_V V_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & a & -b \\ 0 & b & a \end{bmatrix} V_3 \tag{13}$$

$$V_{3x} - U_{3x} = 0 \tag{14a}$$

$$V_{3y}a - U_{3y} - V_{3z}b = 0 \tag{14b}$$

$$V_{3z}a - U_{3z} - V_{3y}b = 0 \tag{14c}$$

where a and b are sine and cosine of the rotation angle. $U_3 = [U_{3x}\ U_{3y}\ U_{3z}]^T$ and $V_3 = [V_{3x}\ V_{3y}\ V_{3z}]^T$. We do variable elimination on Equations 14b and 14c to solve for a which can be back-substituted to solve for b. The full expressions for a and b are

$$a = \frac{U_{3y}V_{3y} + U_{3z}V_{3z}}{V_{3y}^2 + V_{3z}^2},\ b = \frac{-U_{3y} + V_{3y}a}{V_{3z}} \tag{15}$$

Finally, the full transformation R and t is given by

$$R = R_N^T R_V R_M,\ t = -RP_1 + Q_1 \tag{16}$$

5 Robust Estimation

Outlier 2D-3D point correspondences are rejected from our proposed method using RANSAC [5]. We compute the reprojection errors of all the 2D-3D point correspondences based on the hypotheses generated from random sets of unique 3-point correspondences. The hypothesis that yields the highest inlier count, i.e. highest number of 2D-3D point correspondences with the respective reprojection error lower than a given threshold, is chosen as the correct solution. As defined in [5], the number of RANSAC iterations needed is given by $\eta = \frac{\ln(1-p)}{\ln(1-w^n)}$, where p is the probability that all selected correspondences are inliers, w is the probability that any selected correspondence is an inlier and n is the number of correspondences needed for the hypothesis. Assuming that $p = 0.99$ and $w = 0.5$, a total of 35 iterations are needed for our 3-point algorithm, i.e. $n = 3$. In contrast, the linear 6-point algorithm [4] where $n = 6$ would require 293 iterations. The efficiency in having less iterations within RANSAC highlights the importance of using the minimal number of point correspondences.

Each hypothesis generated by RANSAC often give rise to more than one real solution from solving the polynomial equation in Section 4.2. We do additional iterations within RANSAC to check the inlier count for each of these solutions from each hypothesis, where the correct solution gives the highest inlier count. It is therefore desirable to have the minimal solution to keep the number of additional RANSAC iterations low. The number of additional RANSAC iterations for our method is halved compared to Chen's method [2] since our method has a minimal solution up to 8 real solutions while Chen's method yields up to 16 real solutions.

6 Non-Linear Refinement

We further refine the estimated pose R and t by minimizing the total reprojection errors from all the inlier point correspondences found from RANSAC. The cost function is given by

$$\underset{R,t}{\operatorname{argmin}} \sum_i \sum_j ||\pi(P_i, X_j) - \mathbf{x}_{ij}||^2 \tag{17}$$

where $\mathbf{x}_{ij}$ is the 2D image point with X_j as its 3D point correspondence and seen by the i^{th} camera C_i that makes up the multi-camera system. $\pi(.)$ is the camera projection function that projects a 3D point onto the 2D image. P_i is the camera projection matrix given by

$$P_i = K_i[R_{C_i}^T R^T \quad -R_{C_i}^T (R^T t + t_{C_i})] \tag{18}$$

where K_i is the camera intrinsics, R_{C_i} and t_{C_i} are the camera intrinsics as defined in Section 3. The minimization of Equation 17 is done with the Google Ceres solver [1] using the Levenberg-Marquardt algorithm.

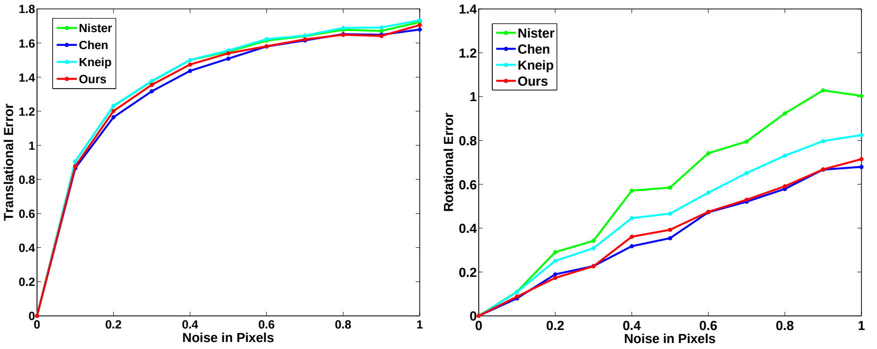
7 Results

We evaluate our proposed multi-camera pose estimation algorithm with both simulations and large-scale real-world datasets.

7.1 Simulations

We compare the accuracy and stability of our algorithm with Nister's [14], Chen's [2] and Kneip's [9] algorithms based on the simulation setup suggested in [17]. The simulated multi-camera system is made up of 4 separate cameras looking front,

[1] http://code.google.com/p/ceres-solver/



(a) (b)

Fig. 5 Average (a) translational and (b) rotational errors from 500 random trials at different pixel noise levels using Nister's [14], Chen's [2], Kneip's [9] and our algorithms. Note that a large part of the translational error for Nister's method (green line) is hidden behind the translational error for Kneip's method (cyan line).

right, left and right with no overlapping field-of-views. Note that the chosen camera configuration and simulated rays are free from the parallel ray degeneracy mentioned in Section 4.3. The absolute orientation used in Chen's method is from [6] while the minimal solution proposed in Section 4.4 is used in our method.

We randomly generate a ground truth camera pose within a given range of [-1 1] m for (x,y,z) and [-0.1 0.1] rad for all angles, i.e. roll, pitch and yaw. We also randomly generate three 3D world points within a given range of [-10 10] m for (x,y,z). The image coordinates are found by reprojecting the 3D points into the respective camera where it is visible. We corrupt the image coordinates with noise ranging from 0.1 to 1 pixel with a 0.1 pixel interval. The pose of the camera in the world frame is computed based on the corrupted image coordinates using the four algorithms. Following [17], we compute the relative translational error as $2||t_{est} - t_{gt}||/(||t_{est}|| + ||t_{gt}||)$ where t_{est} and t_{gt} are the estimated and ground truth translations. The relative rotational error is computed as the norm of the Euler angles from $R_{est}R_{gt}^T$ where R_{est} and R_{gt} are the estimated and ground truth rotation matrices.

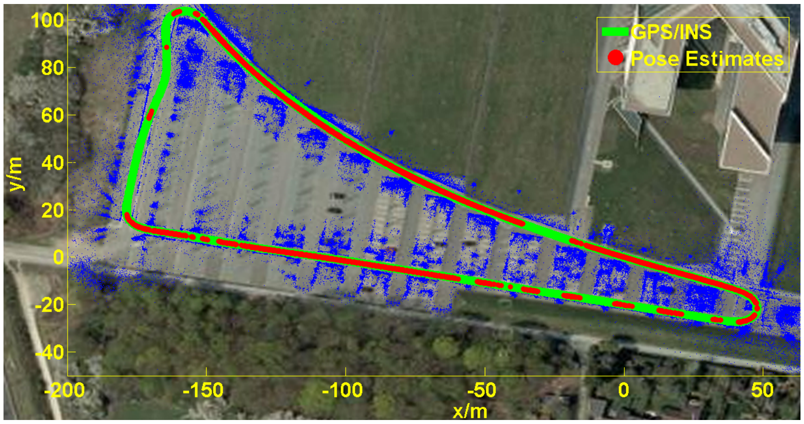
Figures 5(a) and 5(b) shows the plots of the average relative translational and rotational errors from 500 random trials per image coordinate noise level. It can be seen that Chen's and our algorithms produced very similar errors for all noise levels. The errors from both Chen's and our algorithms are also significantly lower than Nister's and Kneip's algorithms. The results imply that the two-step approaches, i.e. Chen's and our algorithms, that solves for the depths and absolute orientation are less susceptible to the influences of noise compared to Nister's direct approach and Kneip's Gröbner basis method.

Time Efficiency of Minimal Solution for Absolute Orientation: We compare the time efficiency of our minimal solution for absolute orientation proposed in Section 4.4 with the standard approach that requires SVD [6, 8]. We randomly generate 500 camera poses in the world frame. For each of these poses, we randomly generate 3 points in the world frame and compute the coordinates of these points in the camera frame. The rigid transformation between the camera and world frames is computed with both approaches. Note that the poses estimated from both methods are always the same as the groundtruth and there is no need for RANSAC in this comparison since the points are noise-free. We obtain the time needed for each trial with the respective method and compute the efficiency of our minimal solution for absolute orientation method over the SVD method as the ratio of the mean time taken by our method to the mean time taken by the SVD method for all the 500 trials. The efficiency ratio is found to be 1.23 and this means that on the average our proposed method is 1.23 times faster than the standard SVD approach.

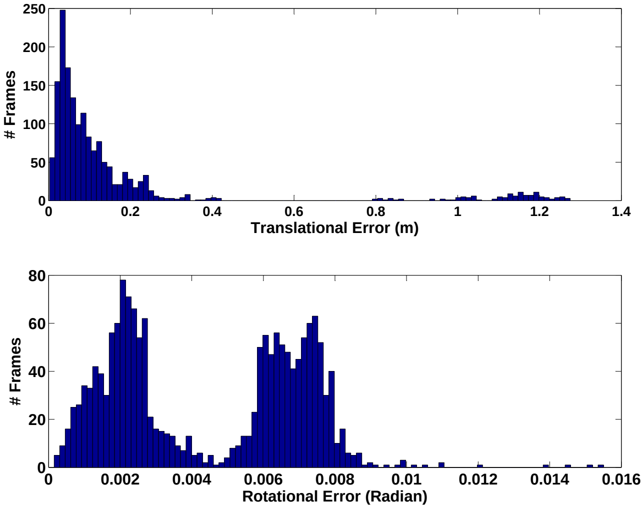
7.2 Real Datasets

Figure 1(a) shows our car platform with 4 fish-eye cameras looking front, rear, left and right with minimal overlapping field-of-views used to collect the datasets for testing our algorithm. The GPS/INS system is also available for ground truth. Figure 1(b) shows 4 sample images from the respective cameras. Figures 6(a) and 7(a) shows two areas for testing our algorithm. TestArea01 and TestArea02 are car parks besides an office building and a supermarket, and covers an area of approximately 140×280m and 160×150m respectively. We collect two datasets separately from each of the test area, i.e. 2×2 datasets - one for building a map and the other for testing our pose estimation algorithm on the map in each test area. To build the maps, we extract the SURF [1] features, and triangulate the 3D points based on the GPS/INS readings. We apply bundle adjustment (implemented with Google Ceres solver) on the GPS/INS poses and triangulated 3D points to get the final maps. The maps also contains all the 2D-3D correspondences of the SURF and 3D points. The blue dots on Figures 6(a) and 7(a) are the 3D points from the maps after bundle adjustment.

The green trajectories on Figures 6(a) and 7(a) are the GPS/INS ground truth readings from the second datasets for testing our pose estimation algorithm on both areas. A total of 2500 and 2100 frames are used for testing. We first create a vocabulary-tree [15] with all the SURF features from the map. For every frame from the test dataset, we extract the SURF features, and query for the frame from the map with the highest similarity score with the vocabulary-tree. We obtain the 2D-3D correspondences of the test and map frames by matching the SURF features. Finally, we compute the pose of the test frame in the map with our multi-camera pose estimation algorithm. Note that a frame refers to a set of 4 images from all the cameras. The red dots on Figures 6(a) and 7(a) are the estimated poses with our algorithm with at least 20 2D-3D correspondences. It can be seen that the poses estimated from our algorithm follows the GPS/INS ground truth closely. Figures 6(b) and 7(b) show the distributions of the translational and rotational errors. We can see that the error distributions are sufficiently low. The translational error is computed as $||t_{est} - t_{gt}||$ where t_{est} and t_{gt} are the translations from the pose estimation and GPS/INS ground truth. The rotational error is computed as the norm of the Eu-



(a)



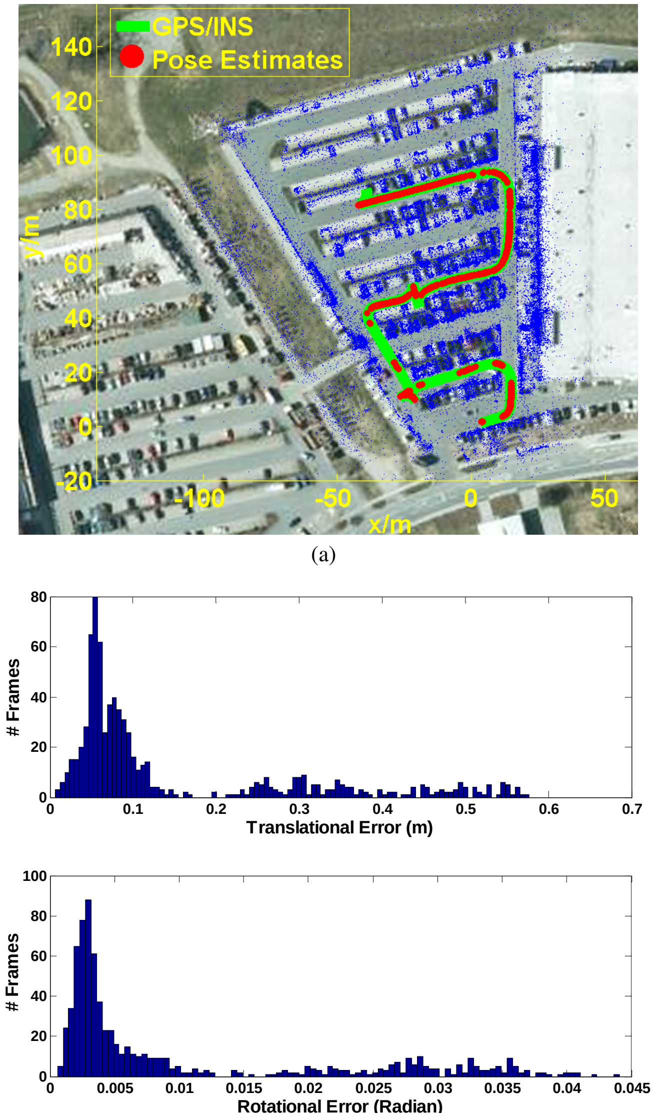
(b)

Fig. 6 (a) Localization results for TestArea01. Results from frames with < 20 correspondences are discarded. (b) Plots showing the distribution of the translational and rotational errors against GPS/INS ground truths.

ler angles from $R_{est}R_{gt}^{T}$ where R_{est} and R_{gt} are the rotation matrices from the pose estimation and GPS/INS ground truth.

8 Conclusion

We showed a new formulation to solve the pose estimation problem of a multi-camera system. Our formulation is intuitive and easy to implement. It is based on the Plücker line coordinates which solves the pose estimation problem in two steps



(a)

(b)

Fig. 7 (a) Localization results for TestArea02. Results from frames with < 20 correspondences are discarded. (b) Plots showing the distribution of the translational and rotational errors against GPS/INS ground truths.

- (a) solve for the depth and (b) solve for the rigid transformation with absolute orientation. We showed that the depths can be solved with a minimal number of 3-point correspondences and leads to an 8 degree polynomial minimal solution. We identified a degenerated case for our method in the case of orthographic projection. We also derived an efficient analytical closed-form minimal solution for the absolute orientation. Our method is verified with both simulations and large-scale real-world datasets from a robotic car platform.

9 Acknowledgement

This work is supported in part by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant #269916 (v-charge) and 4DVideo ERC Starting Grant Nr. 210806.

References

1. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). In *Computer Vision and Image Understanding*, volume 110, pages 346–359, June 2008.
2. C. S. Chen and W. Y. Chang. On pose recovery for generalized visual sensors. In *Pattern Analysis and Machine Intelligence*, volume 26, pages 848–861, 2004.
3. D. A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.).* Springer, 1997.
4. A. Ess, A. Neubeck, and L. Van Gool. Generalised linear pose estimation. In *British Machine Vision Conference*, September 2007.
5. M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, pages 381–395, June 1981.
6. R.M. Haralick, D. Lee, K. Ottenburg, and M. Nolle. Analysis and solutions of the three point perspective pose estimation problem. In *Computer Vision and Pattern Recognition*, pages 592–598, 1991.
7. R. I. Hartley and R. Gupta. Linear pushbroom cameras. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 963–975, 1994.
8. B.K.P. Horn. Closed form solutions of absolute orientation using unit quaternions. In *JOSA-A*, volume 4, pages 629–642, 1987.
9. L. Kneip, P. Furgale, and R. Siegwart. Using multi-camera systems in robotics: Efficient solutions to the npnp problem. In *International Conference on Robotics and Automation*, 2013.
10. G. H. Lee, F. Fraundorfer, and M. Pollefeys. Motion estimation for a self-driving car with a generalized camera. In *Computer Vision and Pattern Recognition*, May 2013.
11. G. H. Lee, F. Fraundorfer, and M. Pollefeys. Structureless pose-graph loop-closures with a multi-camera system on a self-driving car. In *International Conference on Intelligent Robots and Systems*, 2013.
12. H. D. Li, R. Hartley, and J. H. Kim. A linear approach to motion estimation using generalized camera models. In *Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
13. F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o(n) solution to the pnp problem. In *International Conference on Computer Vision*, pages 1–8, 2007.
14. D. Nistér. A minimal solution to the generalised 3-point pose problem. In *Computer Vision and Pattern Recognition*, volume 1, pages 560–567, 2004.
15. D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.
16. R. Pless. Using many cameras as one. In *Computer Vision and Pattern Recognition*, volume 2, pages 587–93, June 2003.
17. L. Quan and Z. D. Lan. Linear n-point camera pose determination. In *Pattern Analysis and Machine Intelligence*, volume 21, pages 774–780, 1999.
18. G. Schweighofer and A. Pinz. Globally optimal o(n) solution to the pnp problem for general camera models. In *British Machine Vision Conference*, pages 1–10, 2008.
19. S. Tariq and F. Dellaert. A multi-camera 6-dof pose tracker. In *International Symposim on Mixed and Augmented Reality*, pages 296–297, 2004.