Article

The International Journal of Robotics Research
1–12
© The Author(s) 2015
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364914557969
ijr.sagepub.com

Minimal solutions for the multi-camera pose estimation problem

Gim Hee Lee[1], Bo Li[2], Marc Pollefeys[3] and Friedrich Fraundorfer[4]

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 three-point correspondences. We show that the minimal solution for the depth of the points along the Plücker lines is an eight-degree polynomial that gives up to eight 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 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 nonlinear refinement step. We verify our approach by showing comparisons with other existing approaches and results from large-scale real-world datasets.

Keywords

Non-perspective pose estimation, multi-camera system, minimal solutions, localization

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, 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.

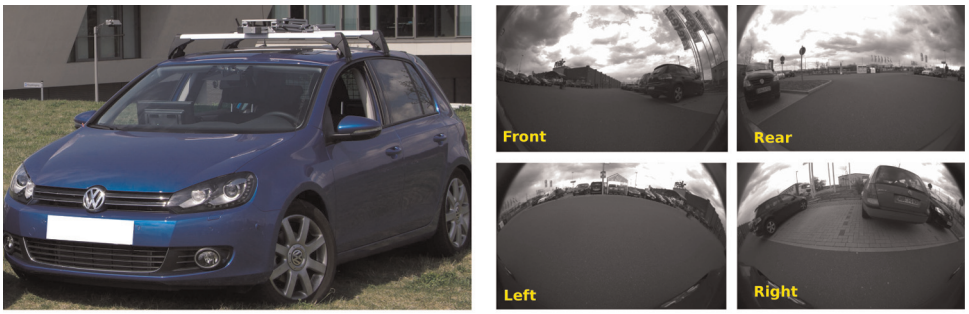
The fact that the light rays from a multi-camera system do not meet at a single center of projection means that all of the classical approaches (Haralick et al., 1991; Quan and Lan, 1999; Moreno-Noguer et al., 2007) for solving the perspective pose problem cannot be used. An alternative approach has 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 random sample consensus (RANSAC; Fischler and Bolles, 1981); see Section 5 for more details.

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 (Pless, 2003; Li et al., 2008; Lee et al., 2013a,b) for relative motion estimation of the multi-camera system. We show that this allows us do a

[1]Mitsubishi Electric Research Laboratories, USA
[2]Baidu Inc., China
[3]Department of Computer Science, ETH Zürich, Switzerland
[4]Faculty of Civil Engineering and Surveying, Technische Universität München, Germany

Corresponding author:
Mitsubishi Electric Research Laboratories, 201 Broadway, 8th Floor, Cambridge, MA 02139, USA.
Email: gimhee.lee@merl.com

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015



(a) (b)

Fig. 1. (a) Our robotic car platform with a multi-camera system made up of four 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 four cameras.

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 (Horn, 1987; Haralick et al., 1991). We show that with a minimal number of three-point correspondences, it leads to an eight-degree polynomial minimal solution that yields up to a maximum of eight 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 singular value decomposition (SVD) and it is inefficient to perform the SVD multiple times to evaluate all of 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 three-point correspondences in closed-form without the need for SVD. Once we have obtained all of the possible solutions for the rigid transformation, we compute the depths of all of 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 details). 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 and Chang (2004) is most related to our method. In this work, they proposed a three-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 eight-degree polynomial that gives up to eight real solutions, which has the advantage of less computational time needed to identify the correct solution. Another drawback of Chen and Chang (2004) 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 (Hartley and Gupta, 1994) (see Section 4.3 for more details). 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.

Nistér (2004) proposed a formulation that directly solves for the rotation and translation parameters. His formulation gives an eight-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 with the single-camera case. Similar to Nistér's method, our method also ends up with an eight-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 by Kneip et al. (2013), the derivation of Nistér's method is not intuitive and requires laborious geometry and algebraic reasonings.

Kneip et al. (2013) presented that most recent work on pose estimation using a multi-camera system. In this work, the authors presented a three-point minimal solution and N-point solution. They first solved for the rotations and point depths with a Gröbner basis (Cox et al., 1997) solver

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

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 by Kneip et al. (2013). Hence, Kneip et al.'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 and Chang's, Nistér's and our methods which solve an eight-degree polynomial that can be done in the order of microseconds as noted by Nistér (2004).

In contrast to the minimal solvers for the pose estimation problem of the multi-camera system, there also exist linear (Ess et al., 2007) and iterative *N*-point (Tariq and Dellaert, 2004; Schweighofer and Pinz, 2008) solutions. The linear solution needs six or more point correspondences and thus is less efficient in RANSAC (Fischler and Bolles, 1981) compared with our minimal solution which requires only three point correspondences. Since that the iterative *N*-point solutions involves computationally expensive iterations, they are usually used to refine the pose estimation after all of 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 (Pless, 2003; Li et al., 2008; Lee et al., 2013a,b, 2014). 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 Pless (2003), Li et al. (2008) and Lee et al. (2013a,b, 2014). 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 the authors' 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 three 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 one single camera (see the perspective case in Section 4.3). We denote the reference frame of the multi-camera system and the world frame as F_G and F_W. The intrinsics and extrinsics of the respective cameras are assumed to be known from calibration (Heng et al., 2013, 2014) 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.

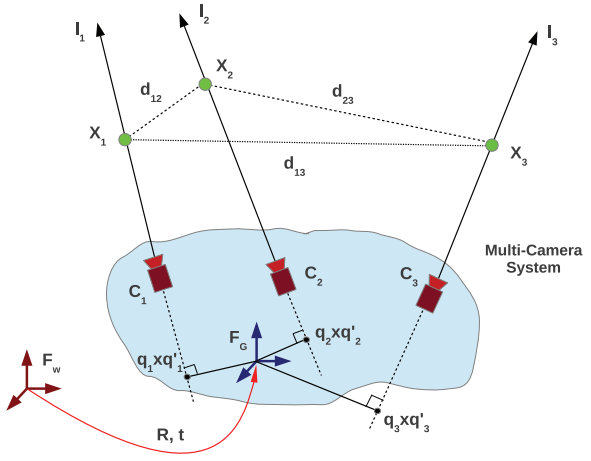


Fig. 2. Illustration of the pose estimation problem for a multi-camera system.

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

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

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 details). Next, we solve for the unknown depths associated with each of the Plücker line using our minimal solution that leads to an eight-degree polynomial giving up to eight real solutions (see Section 4.2 for more details). 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 details).

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

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 Pless (2003), 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 $l_i = [q_i^{\mathrm{T}},\ q'^{\mathrm{T}}_i]^{\mathrm{T}}$ 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 obtain

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

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

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

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 (Cox et al., 1997) 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 eight-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 (Cox et al., 1997) made up of the coefficients. A maximum of up to eight real roots can be obtained for λ_3. As suggested by Nistér (2004), a more efficient way to solve for the roots of the eight-degree polynomial is by using the Sturm sequences.

We can find λ_2 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 of 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 six special cases for the multi-camera pose estimation problem, where the first five special cases are mentioned by Chen and Chang (2004). In particular, we compare the similarities and differences between the existing methods and our method under these six 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 the constraint for Plücker lines 1 and 2 in Equation (4a) becomes

$$\lambda_2 = \lambda_1 + c_{12} \tag{7}$$

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

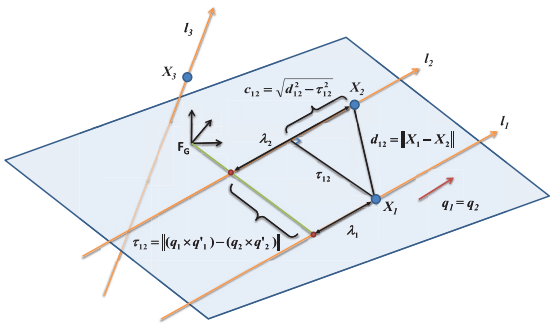


Fig. 4. Illustration of the partially parallel case.

where

$$c_{12}=\sqrt{||X_1-X_2||^2-||(q_1\times q'_1)-(q_2\times q'_2)||^2} \quad (8)$$

is a known value from the Plücker lines $[q_1^{\mathrm{T}},\ q'^{\mathrm{T}}_1]^{\mathrm{T}}$ and $[q_2^{\mathrm{T}},\ q'^{\mathrm{T}}_2]^{\mathrm{T}}$, and distance between the 3D points $(X_1,\ X_2)$. Applying the Sylvester Resultant for variable elimination on the system of polynomial formed by Equations (7), (4b) and (4c), we get a four-degree univariate polynomial minimal solution in terms of λ_3 that can be solved in closed form. A similar four-degree polynomial minimal solution was obtained for Chen and Chang's and Nistér's methods.

Case 2: Perspective. The three light rays pass through a common center of projection in the perspective case, i.e. all of the 2D–3D correspondences are from one single camera in the multi-camera system. Let us choose the camera reference frame F_G to be the center of projection as illustrated in Figure 5. 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), we get the following system of polynomials

$$\kappa_{11}\lambda_1+\kappa_{12}\lambda_1\lambda_2+\kappa_{13}\lambda_2=0 \quad (9a)$$

$$\kappa_{21}\lambda_1+\kappa_{22}\lambda_1\lambda_3+\kappa_{23}\lambda_3=0 \quad (9b)$$

$$\kappa_{31}\lambda_2+\kappa_{32}\lambda_2\lambda_3+\kappa_{33}\lambda_3=0 \quad (9c)$$

where κ are the coefficients made up of the known normalized image coordinates. Applying the Sylvester resultant for variable elimination, we get a four-degree polynomial minimal solution that can be solved in closed-form. This result is similar to Chen and Chang's and Nistér's methods, and the P3P solution for a perspective camera (Haralick et al., 1991). Note that a four-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 as shown in Figure 6. 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 and Chang (2004) to define the distance parameters cannot be used in the case where all of the three rays respectively lie on parallel planes. As a result, an alternative representation has to be made.

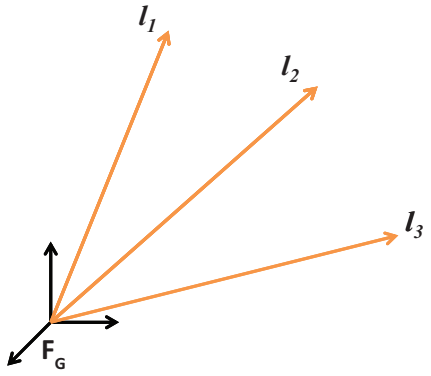


Fig. 5. Illustration of the perspective case.

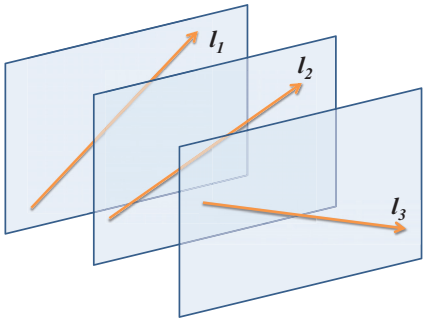


Fig. 6. Illustration of the parallel plane case.

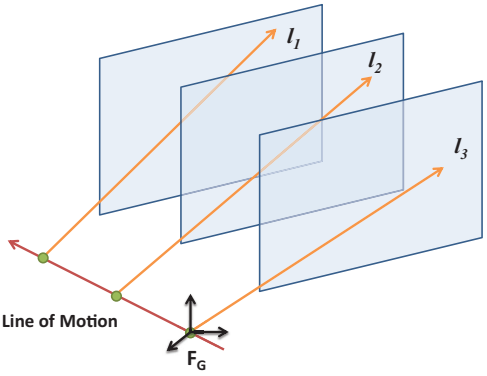


Fig. 7. Illustration of the linear pushbroom case.

Case 4: Linear pushbroom. There is only one camera in the case of linear pushbroom (Hartley and Gupta, 1994). As illustrated in Figure 7, the camera moves through a straight line of motion with a known speed and takes images at

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

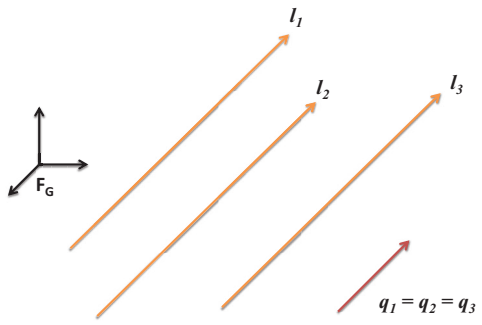


Fig. 8. Illustration of the orthographic case.

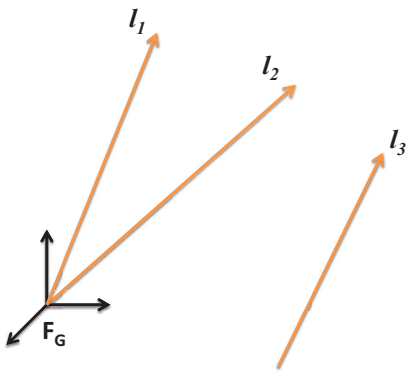


Fig. 9. Illustration of the partially parallel case.

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 and Chang's method.

Case 5: Orthographic. For orthographic projection, all of the light rays are parallel. Hence, all of the unit directions of the Plücker lines are equal, i.e. $q_1 = q_2 = q_3$. Each pair of parallel lines forms a constraint similar to Equation (7) and we get the following system of polynomials:

$$\lambda_2 = \lambda_1 + c_{12} \tag{10a}$$

$$\lambda_3 = \lambda_1 + c_{13} \tag{10b}$$

$$\lambda_3 = \lambda_2 + c_{23} \tag{10c}$$

where c_{12} is defined in Equation (8). Similarly, c_{13} and c_{23} are also known values defined in similar form as Equation (8). An infinite number of solutions exist for λ_1, λ_2 and λ_3 from the system of polynomials. 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 (Chen and Chang, 2004; Nistér, 2004; Tariq and Dellaert, 2004; Ess et al., 2007; Schweighofer and Pinz, 2008; Kneip et al., 2013) on pose estimation for the multi-camera system.

Case 6: Partially perspective. In this case, two of the three light rays pass through a common center of projection, i.e. two of the 2D–3D correspondences are from one single camera in the multi-camera system. An example of the partially perspective case is shown in Figure 9. Let us choose the camera reference frame F_G to be at the center of projection of the two rays with a common center of projection. As a result, the extrinsics become $t_{C_1} = t_{C_2} = 0$, $t_{C_3} \neq 0$, $R_{C_1} = R_{C_2} = I$ and $R_{C_3} \neq I$. Substituting these values into Equation (3), we get the following system of polynomials

$$\kappa_{11}\lambda_1 + \kappa_{12}\lambda_1\lambda_2 + \kappa_{13}\lambda_2 = 0 \tag{11a}$$

$$k_{21}\lambda_1^2 + (k_{22}\lambda_3 + k_{23})\lambda_1 + (k_{24}\lambda_3^2 + k_{25}\lambda_3 + k_{26}) = 0 \tag{11b}$$

$$k_{31}\lambda_2^2 + (k_{32}\lambda_3 + k_{33})\lambda_2 + (k_{34}\lambda_3^2 + k_{35}\lambda_3 + k_{36}) = 0 \tag{11c}$$

where the coefficients κ and k are formed from the Plücker line coordinates. Equation (11a), which is similar to the constraints from the perspective case, is from the two perspective rays. Equations (11b) and (11c) are similar to the general case constraint in Equation (4). We get a eight-degree univariate polynomial in terms of λ_3 after applying the Sylvester Resultant on the system of polynomials. This shows that our method, similar to Chen and Chang (2004) and Nistér (2004), works even when two of the 2D–3D correspondences are from one single camera in the multi-camera system.

4.4. Minimal solution for absolute orientation

Absolute orientation can be solved using the methods from Horn (1987) and Haralick et al. (1991). However, these methods require a computationally inefficient step of SVD that becomes an overhead when it is used numerous times within RANSAC to compute all of the hypothesis solutions. We present a minimal solution that 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

$$P_i = RQ_i + t, \quad i = 1, 2, 3 \tag{12}$$

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.

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

M_2 and N_2. The x-axis of F_M and F_N can be aligned by applying the following transformations

$$M_2 = \begin{bmatrix} M_{2x} \\ M_{2y} \\ M_{2z} \end{bmatrix} = R_M \begin{bmatrix} \|M_2\| \\ 0 \\ 0 \end{bmatrix}, \quad N_2 = \begin{bmatrix} N_{2x} \\ N_{2y} \\ N_{2z} \end{bmatrix} = R_N \begin{bmatrix} \|N_2\| \\ 0 \\ 0 \end{bmatrix} \tag{13}$$

where R_M and R_N are unknown rotation matrices that align the two x-axes. 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

$$R_M = R_{Mz} R_{My} = \begin{bmatrix} ce & -f & de \\ cf & e & df \\ -d & 0 & c \end{bmatrix} \tag{14}$$

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 (14) into Equation (13), we get the following three constraints

$$\|M_2\| ce - M_{2x} = 0 \tag{15a}$$

$$\|M_2\| cf - M_{2y} = 0 \tag{15b}$$

$$-M_{2z} - \|M_2\| d = 0 \tag{15c}$$

where d can be calculated directly from Equation (15c) and c can then be computed with the Pythagoras identity. Here e and f can be solved by substituting c into Equations (15a) and (15b). The full expressions for solving a, b, c and d are given as follows

$$d = \frac{-M_{2z}}{\|M_2\|}, \quad c = \sqrt{1 - d^2}, \quad e = \frac{M_{2x}}{\|M_2\| c}, \quad f = \frac{M_{2y}}{\|M_2\| c} \tag{16}$$

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

$$U_i = R_M^{\mathrm{T}} M_i, \quad V_i = R_N^{\mathrm{T}} N_i, \quad i = 1, 2, 3 \tag{17}$$

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

$$U_3 = R_V V_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & a & -b \\ 0 & b & a \end{bmatrix} V_3 \tag{18}$$

which can be expanded into three independent constraints

$$V_{3x} - U_{3x} = 0 \tag{19a}$$

$$V_{3y} a - U_{3y} - V_{3z} b = 0 \tag{19b}$$

$$V_{3z} a - U_{3z} - V_{3y} b = 0 \tag{19c}$$

where a and b are sine and cosine of the rotation angle. Here $U_3 = [U_{3x} U_{3y} U_{3z}]^{\mathrm{T}}$ and $V_3 = [V_{3x} V_{3y} V_{3z}]^{\mathrm{T}}$. We do variable elimination on Equations (19b) and (19c) to solve for a which can be back-substituted to solve for b. The full expressions for a and b are

$$a = \frac{U_{3y} V_{3y} + U_{3z} V_{3z}}{V_{3y}^2 + V_{3z}^2} \tag{20a}$$

$$b = \frac{-U_{3y} + V_{3y} a}{V_{3z}} \tag{20b}$$

In comparison with the methods proposed by Horn (1987) and Haralick et al. (1991), our method does not enforce orthogonality in the rotation matrix. Hence, in the presence of noise, the constraints from Equation (19) cannot be satisfied and the orthogonality property of the rotation matrix R_V is lost. Since $a = \cos\theta$ and $b = \sin\theta$, where θ is the rotation angle around the x-axis, we enforce orthogonality on R_V by making

$$a \leftarrow \begin{cases} \min(1, a), & \text{if } a \geq 0 \\ \max(-1, a), & \text{if } a < 0 \end{cases} \tag{21}$$

and

$$b \leftarrow \begin{cases} |\sin(\cos^{-1}(a))|, & \text{if } b \geq 0 \\ -|\sin(\cos^{-1}(a))|, & \text{if } b < 0 \end{cases} \tag{22}$$

Finally, the full transformation R and t is given by

$$R = R_N^{\mathrm{T}} R_V R_M, \quad t = -RP_1 + Q_1 \tag{23}$$

It is important to note that our method trade-off robustness for efficiency, i.e. our method without using SVD is faster but it is not a least-squares estimate. This means that the accuracy from our method deteriorates in the presence of very high noise. Nonetheless, our simulation results in Section 7.1 show that the accuracy of our method is comparable with Horn (1987) up to 1 pixel noise in the 2D image features. A 1 pixel noise is usually the upper bound for many image feature detectors such as speeded-up robust features (SURF; Bay et al., 2008). Moreover, we shall see in the next section that image features with very high noise are removed as outliers in the RANSAC process.

5. Robust estimation

Outlier 2D–3D point correspondences are rejected from our proposed method using RANSAC (Fischler and Bolles, 1981). We compute the reprojection errors of all of the 2D–3D point correspondences based on the hypotheses generated from random sets of unique three-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 by Fischler and Bolles (1981), the number of RANSAC iterations needed is given by $\eta = \frac{\ln(1-p)}{\ln(1-w^n)}$, where

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

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 three-point algorithm, i.e. $n = 3$. In contrast, the linear six-point algorithm (Ess et al., 2007) 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 with the method of Chen and Chang (2004) since our method has a minimal solution up to 8 real solutions while Chen and Chang's method yields up to 16 real solutions.

6. Nonlinear refinement

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

$$\underset{R,t}{\operatorname{argmin}} \sum_i \sum_j \left\| \pi(P_i, X_j) - \mathbf{x}_{ij} \right\|^2 \tag{24}$$

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

$$P_i = K_i[R_{C_i}^{\mathrm{T}} R^{\mathrm{T}} - R_{C_i}^{\mathrm{T}}(R^{\mathrm{T}} t + t_{C_i})] \tag{25}$$

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 (24) is done with the Google Ceres solver (see https://code.google.com/p/ceres-solver/) 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

7.1.1. Accuracy and stability comparisons. We compare the accuracy and stability of our algorithm with those of Nistér (2004), Chen and Chang (2004) and Kneip et al. (2013) based on the simulation setup suggested by Quan and Lan (1999). The simulated multi-camera system is made up of four separate cameras looking front, rear, left and right with no overlapping fields of view. 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 and Chang's method is from Haralick et al. (1991) while the minimal solution proposed in Section 4.4 is used in our method. To make a fair comparisons of the accuracy and stability of the algorithms, we do not apply RANSAC and nonlinear refinement in the simulations.

We randomly generate a ground truth camera pose within a given range of $[-3\ 3]$ 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 $[-50\ 50]$ 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 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 Quan and Lan (1999), we compute the relative translational error as $2||t_{\mathrm{est}} - t_{\mathrm{gt}}||/(||t_{\mathrm{est}}|| + ||t_{\mathrm{gt}}||)$ 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 $R_{\mathrm{est}}R_{\mathrm{gt}}^{\mathrm{T}}$ where R_{est} and R_{gt} are the estimated and ground truth rotation matrices.

Figures 10(a) and 10(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 the errors from both our algorithm and that of Chen and Chang (2004) are lower than the algorithms of Nistér (2004) and Kneip et al. (2013). The results imply that the two-step approaches, i.e. Chen and Chang's and our algorithms, that solves for the depths and absolute orientation are less susceptible to the influences of noise compared with Nistér's direct approach and Kneip et al.'s Gröbner basis method. It was mentioned in Section 4.4 that our minimal solution for absolute orientation is not a least-squares solution and therefore less robust to noise. Nonetheless, from the simulation results, we observe that the estimation errors from our algorithm is only marginally higher than Chen and Chang's algorithm that used the absolute orientation from Haralick et al. (1991). The estimation errors from our algorithm also remain relatively low with increasing pixel noise.

7.1.2. Time efficiency of the 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 (Horn, 1987; Haralick et al., 1991). Figure 11 shows the error bar (means and standard deviations) plot of the running times needed for our minimal solution for absolute orientation

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

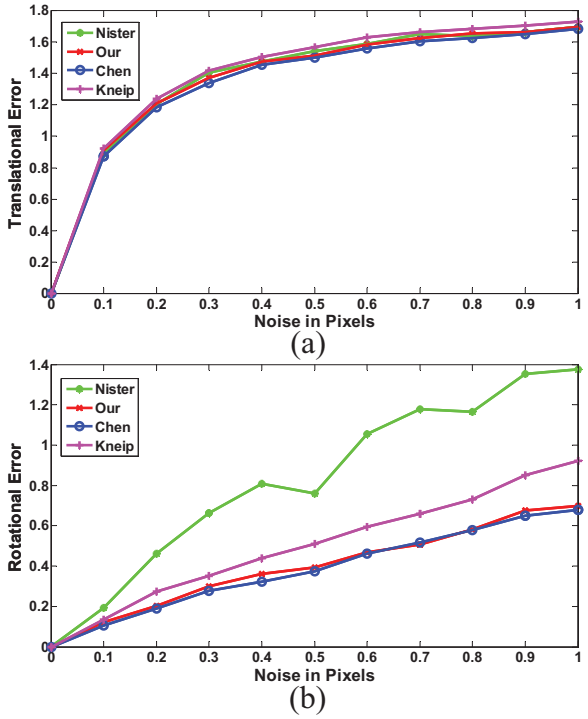


Fig. 10. Average (a) translational (no units) and (b) rotational (radians) errors from 500 random trials at different pixel noise levels using our algorithm and those of Nistér (2004), Chen and Chang (2004) and Kneip et al. (2013). Note that a large part of the translational error for Nistér's method is hidden behind the translational error for Kneip et al.'s method.

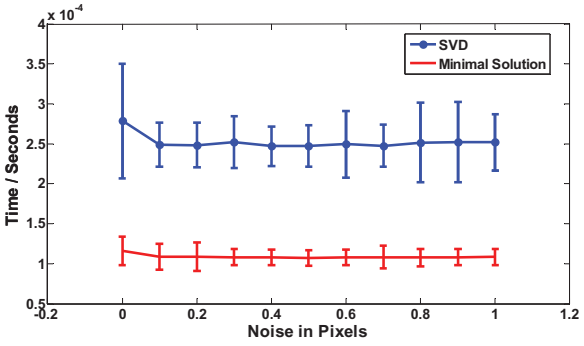


Fig. 11. Comparison of the running times from our absolute orientation minimal solution and Haralick et al. (1991).

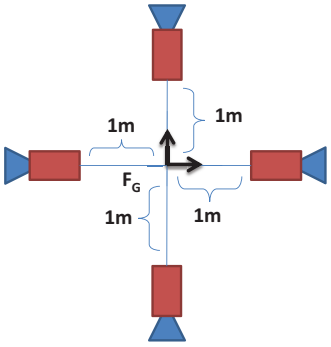


Fig. 12. Simulated camera setup for studying the effects of calibration errors.

and the SVD approach proposed by Haralick et al. (1991) and Horn (1987) over pixel noises in the range of 0 to 1 pixel at an interval of 0.1 pixel. Similar to the accuracy and stability comparison simulations, for each pixel noise, we randomly generate 500 ground truth camera pose within a given range of [−3 3] 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 [−50 50] m for (x, y, z). The times recorded on Figure 11 are the mean (with standard deviations) times taken from the minimal and SVD absolute orientation, respectively, after the computation of the depths of the points along the Plücker lines using the method proposed in Section 4 for all the 500 trials under each pixel noise. It can be seen from the error bar plot in Figure 11 that the computation with our minimal solution for absolute orientation is on the average about 2.5 times faster than the standard SVD approach. It is also interesting to note that the standard deviations of the running time from our minimal solution is smaller than the SVD approach.

7.1.3. Effects of calibration errors. We study the effects of extrinsics calibration errors on our proposed minimal solutions for multi-camera pose estimation in simulations. Again, similar to the accuracy and stability comparison simulations, we study the effects of calibration errors with a simulated multi-camera system that is made up of four cameras looking front, rear, left and right. Each of the cameras is perfectly aligned in its respective viewing direction and fixed at 1 m away from the multi-camera reference frame as illustrated in Figure 12. We simulate random camera poses within a given range of [−3 3] m for (x, y, z) and [−0.1 0.1] rad for all angles, and 3D world points within a given range of [−50 50] m. For each set of randomly simulated camera pose, we project the 3D world points onto the camera image to get the correspondent 2D image coordinates. The 2D image coordinates and the 2D–3D correspondences are kept fixed during the simulation. We check the effects of the calibration errors in the translation and rotation components separately.

Figure 13(a) shows a plot of the mean translation and rotation errors in the pose estimation from 500 trials per calibration error over a range of up to 10 cm at an interval of 1 cm in all translation components, i.e. (x, y, z). No rotation calibration errors are added. The translation and rotation errors are the norm of the respective relative translation and rotation (Euler angles) between the estimated pose and ground truth. Figure 13(b) shows the plot of the mean translation and rotation errors in the pose estimation from 500 trials per calibration error over a range of up to 1° at an interval of 0.1° in all rotation components, i.e. Euler angles. No translation calibration errors are added. It can be

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

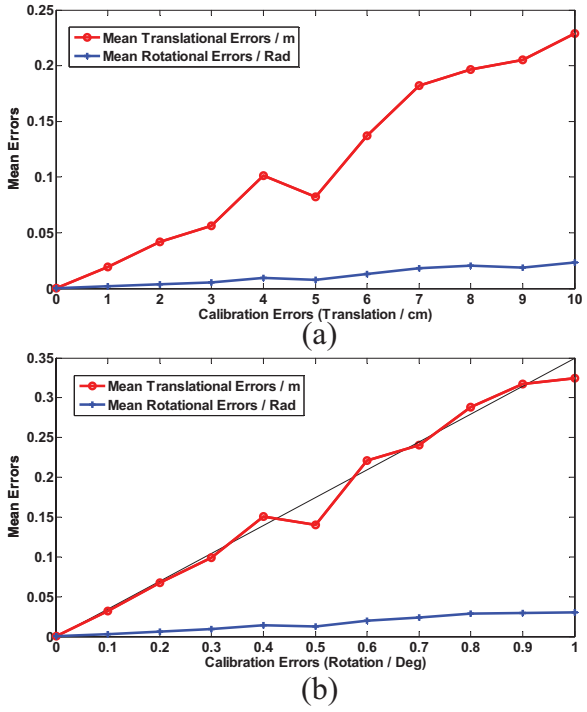


Fig. 13. Pose estimation errors in the presence of (a) translation and (b) rotation calibration errors.

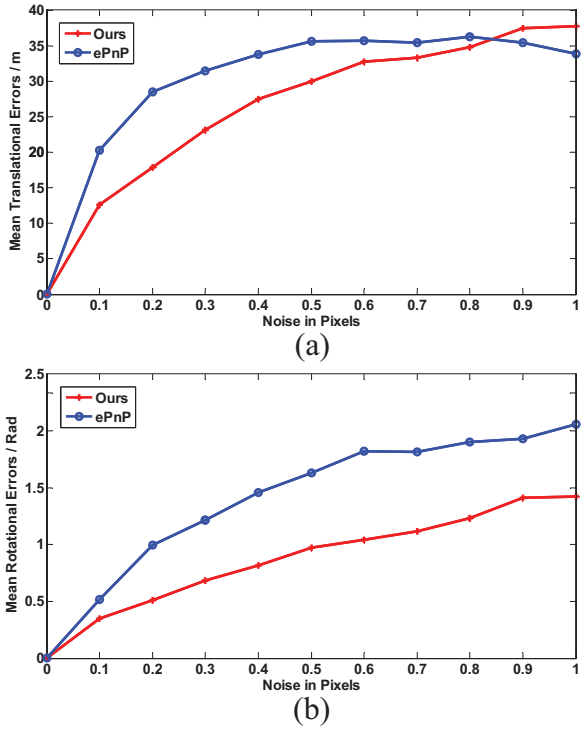


Fig. 14. Mean (a) translational and (b) rotational errors from our method and Moreno-Noguer et al. (2007) for the perspective case.

observed that the translation and rotation errors are sufficiently low for both translation and rotation calibration errors. The pose estimation errors are also slightly higher over the 1° rotation calibration error interval compared with the 10 cm translation calibration error interval.

7.1.4. Perspective case comparisons. The special case of perspective rays mentioned in Section 4.3 is important because it corresponds to the perspective pose estimation problem of a single camera. We compare the accuracy of our algorithm in this degenerate case with the perspective pose estimation algorithm for a single camera from Moreno-Noguer et al. (2007). Figures 14(a) and 14(b) show the mean translation and rotation errors from 500 trials per pixel noise over the range of 0 to 1 pixel at an interval of 0.1 pixel. It can be observed that the mean translation and rotation errors from our method is consistently lower than the pose estimation algorithm for a single camera (Moreno-Noguer et al., 2007).

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 fields of view used to collect the datasets for testing our algorithm. The GPS/INS system is also available for ground truth. Figure 1(b) shows four sample images from the respective cameras. Figures 15(a) and 16(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 m $\times$ 280 m and 160 m $\times$ 150 m, respectively. We collect two datasets separately from each of the test area, i.e. 2 $\times$ 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 (Bay et al., 2008) 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 of the 2D–3D correspondences of the SURF and 3D points. The blue dots on Figures 15(a) and 16(a) are the 3D points from the maps after bundle adjustment.

The green trajectories in Figures 15(a) and 16(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 (Nistér and Stewénius, 2006) 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 four images from all of the cameras. The red dots in Figures 15(a) and 16(a) are the estimated poses with our algorithm with at least 20 2D–3D correspondences. An average of 60 correspondences are

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

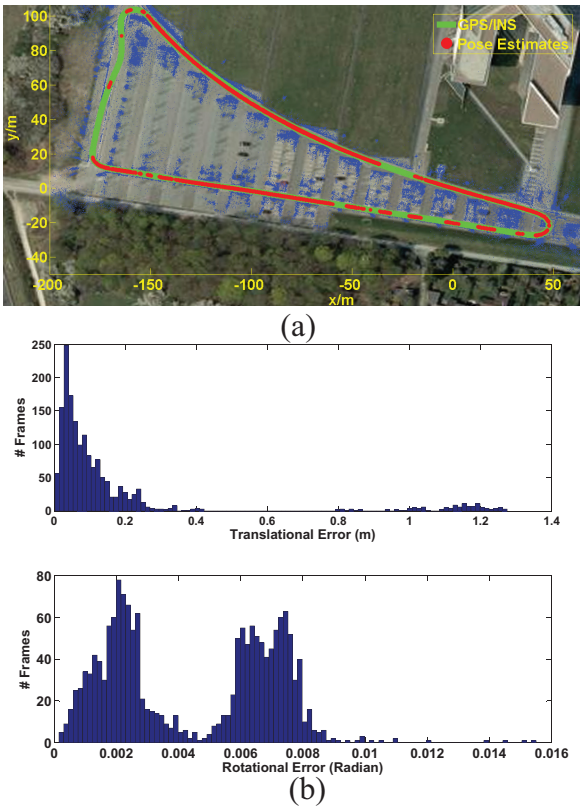


Fig. 15. (a) Localization results for TestArea01. Results from frames with < 20 correspondences are discarded, i.e. regions where estimated poses are not shown. (b) Plots showing the distribution of the translational and rotational errors against GPS/INS ground truths.

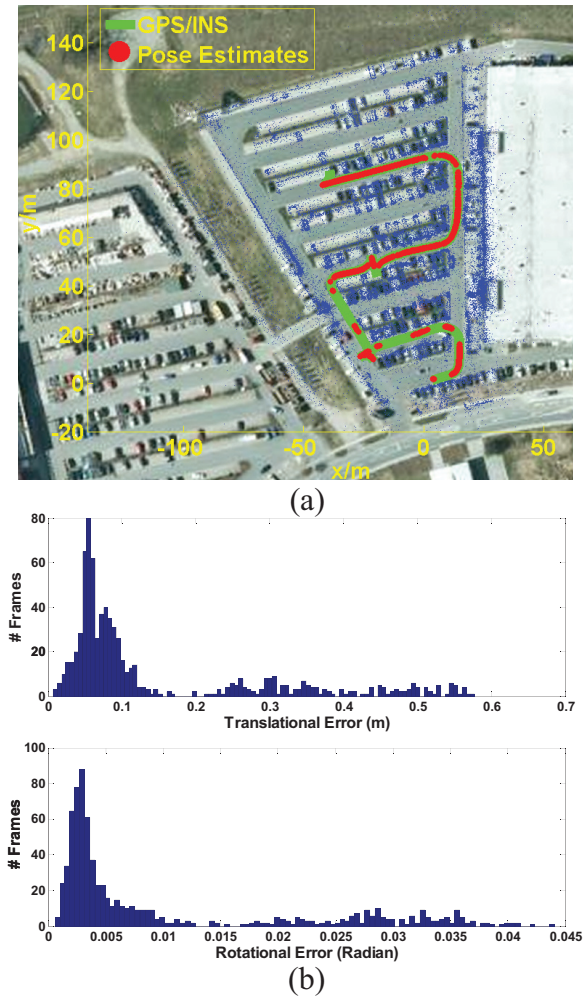


Fig. 16. (a) Localization results for TestArea02. Results from frames with < 20 correspondences are discarded, i.e. regions where estimated poses are not shown. (b) Plots showing the distribution of the translational and rotational errors against GPS/INS ground truths.

found between each camera image and the map for both datasets. It can be seen that the poses estimated from our algorithm follows the GPS/INS ground truth closely. Figures 15(b) and 16(b) show the distributions of the translational and rotational errors. We can see that the error distributions are sufficiently low. We observe that the pose estimates with higher errors are from images with number of correspondences closer to the threshold, i.e. 20. It is also important to note that we make the assumption that the GPS/INS reference frame is identical to the multi-camera reference frame. The bi-modal distribution from the rotation error in Figure 15(b) is probably due to the slight imprecision of this assumption. The translational error is computed as $||t_{\text{est}} - t_{\text{gt}}||$ 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 Euler angles from $R_{\text{est}}R_{\text{gt}}^{\text{T}}$ 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) 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 three-point correspondences and leads to an eight-degree polynomial minimal solution. We identified a degenerate 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.

Acknowledgements

A large part of this work was done when the first and second authors were in the Department of Computer Science, ETH Zürich, Switzerland.

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015

Funding

This work was supported in part by the European Community (grant number 269916; v-charge) and 4DVideo (ERC Starting Grant number 210806).

References

Bay H, Ess A, Tuytelaars T and Van Gool L (2008) Speeded-up robust features (SURF). *Computer Vision and Image Understanding* 110: 346–359.

Chen CS and Chang WY (2004) On pose recovery for generalized visual sensors. *Pattern Analysis and Machine Intelligence* 26: 848–861.

Cox DA, Little J and O'Shea D (1997) *Ideals, Varieties, and Algorithms - An Introduction to Computational Algebraic Geometry and Commutative Algebra,* 2nd edn. New York: Springer.

Ess A, Neubeck A and Van Gool L (2007) Generalised linear pose estimation. In: *British Machine Vision Conference*.

Fischler MA and Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24: 381–395.

Haralick R, Lee D, Ottenburg K and Nolle M (1991) Analysis and solutions of the three point perspective pose estimation problem. In: *Proceedings (CVPR'91) IEEE computer society conference on computer vision and pattern recognition*, pp. 592–598.

Hartley RI and Gupta R (1994) Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19: 963–975.

Heng L, Bürki M, Lee GH, Furgale P, Siegwart R and Pollefeys M (2014) Infrastructure-based calibration of a multi-camera rig. In: *IEEE international conference on robotics and automation (ICRA)*.

Heng L, Li B and Pollefeys M (2013) Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*.

Horn B (1987) Closed form solutions of absolute orientation using unit quaternions. *JOSA A* 4: 629–642.

Kneip L, Furgale P and Siegwart R (2013) Using multi-camera systems in robotics: Efficient solutions to the npnp problem. In: *International conference on robotics and automation*.

Lee GH, Fraundorfer F and Pollefeys M (2013a) Motion estimation for a self-driving car with a generalized camera. In: *Computer Vision and Pattern Recognition*.

Lee GH, Fraundorfer F and Pollefeys M (2013b) Structureless pose-graph loop-closures with a multi-camera system on a self-driving car. In: *International conference on intelligent robots and systems*.

Lee GH, Pollefeys M and Fraundorfer F (2014) Relative pose estimation for a multi-camera system with known vertical direction. In: *IEEE conference on computer vision and pattern recognition (CVPR)*.

Li HD, Hartley R and Kim JH (2008) A linear approach to motion estimation using generalized camera models. In: *IEEE Conference on computer vision and pattern recognition, 2008 (CVPR 2008)*, pp. 1–8.

Moreno-Noguer F, Lepetit V and Fua P (2007) Accurate non-iterative O(*N*) solution to the PNP problem. In: *International conference on computer vision*, pp. 1–8.

Nistér D (2004) A minimal solution to the generalised 3-point pose problem. In: *IEEE conference on computer vision and pattern recognition, 2004 (CVPR 2004)*, vol. 1, pp. 560–567.

Nistér D and Stewénius H (2006) Scalable recognition with a vocabulary tree. In: *IEEE conference on computer vision and pattern recognition, 2006 (CVPR 2006)*, pp. 2161–2168.

Pless R (2003) Using many cameras as one. In: *IEEE conference on computer vision and pattern recognition, 2003 (CVPR 2003)*, vol. 2, pp. 587–593.

Quan L and Lan ZD (1999) Linear *n*-point camera pose determination. *Pattern Analysis and Machine Intelligence* 21: 774–780.

Schweighofer G and Pinz A (2008) Globally optimal O(*N*) solution to the PNP problem for general camera models. In: *British machine vision conference*, pp. 1–10.

Tariq S and Dellaert F (2004) A multi-camera 6-DOF pose tracker. In: *International symposim on mixed and augmented reality*, pp. 296–297.

Downloaded from ijr.sagepub.com at DEUTSCHES ZENTRUM FUER LUFT on December 6, 2015