

Relative Pose Estimation for a Multi-Camera System with Known Vertical Direction

Gim Hee Lee¹, Marc Pollefeys¹, and Friedrich Fraundorfer²

Department of Computer Science¹
ETH Zürich, Switzerland

{glee@student, pomarc@inf}.ethz.ch

Faculty of Civil Engineering and Surveying²
Technische Universität München, Germany

friedrich.fraundorfer@tum.de

Abstract

In this paper, we present our minimal 4-point and linear 8-point algorithms to estimate the relative pose of a multi-camera system with known vertical directions, i.e. known absolute roll and pitch angles. We solve the minimal 4-point algorithm with the hidden variable resultant method and show that it leads to an 8-degree univariate polynomial that gives up to 8 real solutions. We identify a degenerated case from the linear 8-point algorithm when it is solved with the standard Singular Value Decomposition (SVD) method and adopt a simple alternative solution which is easy to implement. We show that our proposed algorithms can be efficiently used within RANSAC for robust estimation. We evaluate the accuracy of our proposed algorithms by comparisons with various existing algorithms for the multi-camera system on simulations and show the feasibility of our proposed algorithms with results from multiple real-world datasets.

1. Introduction

A multi-camera system refers to a system of cameras that are rigidly fixed onto a single body and it offers several advantages for ego-motion estimation. Since the multi-camera system is made up of individual cameras, it can be set in a configuration that maximizes the field-of-view. A maximal field-of-view of the environment means that more image features can be robustly detected and tracked, which is the most fundamental but crucial step in ego-motion estimation. One other major advantage of the multi-camera system is the fact that the metric scale can be directly obtained from the epipolar geometry as described in earlier works [13, 16, 11]. This means that as compared to a stereo camera, there is greater flexibility in the configuration of the multi-camera system because there is no need for large or any overlapping field-of-view to retrieve the metric scale. Figure 1 shows an example of our multi-camera system made up of four fish-eye cameras fixed onto a car for ego-

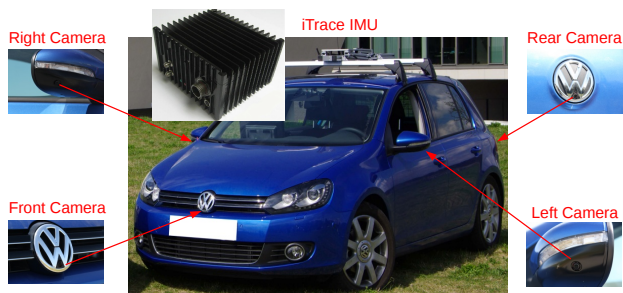


Figure 1. An example of a multi-camera system with known vertical from the IMU on a car.

motion estimation.

The main difference between the multi-camera system and the standard pinhole camera is the absence of a single center of projection. The light rays that passed through the multi-camera system do not meet at a single center of projection, i.e. non-central projection. To overcome this problem, Pless [13] proposed to express the light rays as Plücker lines and requires 17-point correspondences to compute the relative motion linearly. In a later work [16], Sturm also came up with similar formulations. In [11], Li *et al.* built on the work of Pless by identifying several cases of degeneracies and proposed the linear 16-point and 14-point algorithms to overcome the degeneracies. The high number of correspondences needed for these algorithms however made them impractical for robust estimation with RANSAC [3] in real-world applications for ego-motion estimation. The minimal 6-point algorithm [15] proposed by Stewénius *et al.* requires much lesser point correspondences but yields up to 64 solutions, which makes the process of identifying the correct solution difficult and computationally expensive.

More recently, Lee *et al.* [9] made use of the fact that their multi-camera system is fixed onto a car to constrain the relative motion with the Ackermann motion model, which requires only 2-point correspondences and gives up to 6 solutions. As a result, it became possible for them to use RANSAC for robust estimation and show results on large

real-world datasets. In [10], Lee *et al.* relaxed the Ackermann motion constraint to a planar constraint to allow for the computation of loop-closure constraints. Despite their success in applying the multi-camera system on real-world datasets for ego-motion estimation, their algorithms work on a very strict assumption that the ground has to be planar. While this assumption is valid for a large part of the roads, there are instances such as ramps at highway entrances or exits, multi-level parking garages and off-road drivings where the assumption would be violated.

In this paper, we make use of the known vertical direction from an Inertial Measurement Unit (IMU) to keep the number of point correspondences low while allowing the full 6 degree-of-freedom for the relative motion to be estimated. While relative [4, 6] and absolute pose [8] estimation with known vertical direction has been shown for a monocular camera, no prior work exists for relative motion estimation with known vertical direction for the multi-camera system. Figure 1 shows our car equipped with a multi-camera system and the iTrace¹ IMU. In particular, we propose the minimal 4-point and linear 8-point algorithms to estimate the relative pose of a multi-camera system with known vertical direction, i.e. known absolute roll and pitch angles. We show that the minimal 4-point algorithm can be solved with the hidden variable resultant [2] that gives up to 8 solutions. We identify a degenerated case in the linear 8-point algorithm when it is solved with the standard SVD approach [5] and adopt a simple strategy from [11] to solve it. We show that our proposed algorithms can be efficiently used within RANSAC for robust estimation. The accuracy of our proposed algorithms is compared with the existing algorithms in simulations and we also demonstrate the feasibility our algorithms on non-planar roads with several real-world datasets taken on a multi-level parking garage.

2. Generalized Epipolar Constraint

In this section, we give a brief description of the multi-camera epipolar constraint, otherwise known as the generalized epipolar constraint, which is needed to follow the derivations of our minimal 4-point and linear 8-point algorithms in the next sections. More details on the generalized epipolar constraint can be found in [13]. The main difference between the generalized camera and a standard pinhole camera is the absence of a single center of projection. To overcome this problem, the light rays from all the cameras in the multi-camera system are expressed as the 6-vector Plücker line with respect to a common arbitrarily chosen reference frame V . Let us denote each camera as C_i and its extrinsics expressed in the reference frame V as $[t_{C_i}, R_{C_i}]$. The intrinsics of each camera is denote by K_{C_i} . The Plücker line that represents the light ray that passes through C_i and

the normalized image point $\hat{x}_{ij} = K_{C_i}^{-1}x_{ij}$ is given by

$$l_{ij} = [u_{ij}^T, (t_{C_i} \times u_{ij})^T]^T \quad (1)$$

where $u_{ij} = R_{C_i}\hat{x}_{ij}$ is the unit direction of the light ray expressed in the reference frame V . As a result, the generalized epipolar constraint can be written as

$$l_{ij}^T \underbrace{\begin{bmatrix} E & R \\ R & 0 \end{bmatrix}}_{E_{GC}} l_{ij} = 0 \quad (2)$$

where $l'_{ij} \leftrightarrow l_{ij}$ are the ray correspondences between two generalized camera frames V' and V . E_{GC} is the 6×6 generalized essential matrix made up of the conventional essential matrix $E = [t]_{\times}R$, where R and t are the rotation and translation between the two frames [5]. The fact that the generalized epipolar constraint is made up the Plücker line correspondences which consist of the extrinsics translation t_{C_i} in metric scale means that the metric scale of the relative translation t between the two frames can be fully determined.

3. Problem Definition

The problem of relative pose estimation for a multi-camera system with known vertical direction is defined as:

Definition 1. *Given a calibrated multi-camera system (i.e. known intrinsics and extrinsics), the vertical directions (roll and pitch angles from the IMU) of two multi-camera system frames and image point correspondences between the two multi-camera system frames, find the relative pose between the two multi-camera system frames.*

4. Minimal 4-Point Algorithm

The pipeline of our minimal 4-point algorithm first transforms the Plücker line correspondences with the roll and pitch angles of the correspondence frames from the IMU. Next, the minimal 4-point algorithm gives the relative pose estimated from the transformed Plücker line correspondences. Lastly, the relative pose in the original correspondence frames is computed.

4.1. Apply Roll and Pitch Angles

The roll and pitch angles from the IMU are given with respect to a reference frame W , where the xy plane is parallel to the ground plane and x -axis aligned with the True North. In a calibrated setting where the relative rotation between the generalized camera frame and IMU is known, let us denote the rotation matrices from the roll and pitch angles of the two correspondence generalized camera frames V' and V with respect to W as $(R'_r, R'_p) \leftrightarrow (R_r, R_p)$. The relative rotation R can now be written as

¹<http://www.imar-navigation.de/index.php/de/>

$$R = R_r'^T R_p'^T \hat{R}_y R_p R_r \quad (3)$$

where $\hat{R}_y = R_y'^T R_y$ and $R_y' \leftrightarrow R_y$ are the rotation matrices from the yaw angles of the two correspondence generalized camera frames V' and V with respect to W . Putting Equation 3 into the generalized essential matrix E_{GC} from Equation 2, we get

$$\begin{bmatrix} [t]_x R_r'^T R_p'^T \hat{R}_y R_p R_r & R_r'^T R_p'^T \hat{R}_y R_p R_r \\ R_r'^T R_p'^T \hat{R}_y R_p R_r & 0 \end{bmatrix} \quad (4)$$

where we can factorize out $R_p R_r$ to the right hand side of the matrix to get

$$\begin{bmatrix} [t]_x R_r'^T R_p'^T \hat{R}_y & R_r'^T R_p'^T \hat{R}_y \\ R_r'^T R_p'^T \hat{R}_y & 0 \end{bmatrix} \begin{bmatrix} R_p R_r & 0 \\ 0 & R_p R_r \end{bmatrix} \quad (5)$$

The generalized essential matrix can be further simplified by replacing $[t]_x R_r'^T R_p'^T$ with $R_r'^T R_p'^T [\hat{t}]_x$, where $R_r'^T R_p'^T$ can be factorized to the left hand side of the matrix to get

$$\begin{bmatrix} R_r'^T R_p'^T & 0 \\ 0 & R_r'^T R_p'^T \end{bmatrix} \underbrace{\begin{bmatrix} [\hat{t}]_x \hat{R}_y & \hat{R}_y \\ \hat{R}_y & 0 \end{bmatrix}}_{\hat{E}_{GC}} \begin{bmatrix} R_p R_r & 0 \\ 0 & R_p R_r \end{bmatrix} \quad (6)$$

\hat{E}_{GC} is the new generalized essential matrix to be solved after factorizing out the roll and pitch rotation matrices. Putting Equation 6 into the generalized epipolar constraint from Equation 2, and dropping the camera and image point indices ij for brevity, we get

$$\underbrace{\left(\begin{bmatrix} R_p' R_r' & 0 \\ 0 & R_p' R_r' \end{bmatrix} l' \right)^T}_{\hat{l}'} \hat{E}_{GC} \underbrace{\left(\begin{bmatrix} R_p R_r & 0 \\ 0 & R_p R_r \end{bmatrix} l \right)}_{\hat{l}} = 0 \quad (7)$$

where $\hat{l}' \leftrightarrow \hat{l}$ are the correspondence Plücker lines after the respective xy plane from the generalized camera frames V' and V are aligned, i.e. the roll and pitch angles are removed.

4.2. Minimal Problem

The task is to solve for \hat{t} and \hat{R}_y after the alignment of the two correspondence generalized camera frames. We write \hat{t} and \hat{R}_y as

$$\hat{t} = \begin{bmatrix} \hat{t}_x \\ \hat{t}_y \\ \hat{t}_z \end{bmatrix}, \quad \hat{R}_y = \frac{1}{1+q^2} \begin{bmatrix} 1-q^2 & -2q & 0 \\ 2q & 1-q^2 & 0 \\ 0 & 0 & 1+q^2 \end{bmatrix} \quad (8)$$

where $q = \tan \frac{\hat{\theta}}{2}$, hence $\cos(\hat{\theta}) = \frac{1-q^2}{1+q^2}$ and $\sin(\hat{\theta}) = \frac{2q}{1+q^2}$ are the trigonometric identities to remove the sines and cosines in the rotation matrix. $\hat{\theta}$ is the yaw angle that make

up \hat{R}_y . Putting the expressions for \hat{t} and \hat{R}_y into the generalized epipolar constraint from Equation 7, we get the following polynomial equation

$$a_1 \hat{t}_x q^2 + a_2 \hat{t}_x q + a_3 \hat{t}_x + a_4 \hat{t}_y q^2 + a_5 \hat{t}_y q + a_6 \hat{t}_y + a_7 \hat{t}_z q^2 + a_8 \hat{t}_z q + a_9 \hat{t}_z + a_{10} q^2 + a_{11} q + a_{12} = 0 \quad (9)$$

where a_1 to a_{12} are the coefficients formed with the Plücker line correspondence $\hat{l}' \leftrightarrow \hat{l}$. We need a minimal of four Plücker line correspondences to solve for the four unknowns $\hat{t}_x, \hat{t}_y, \hat{t}_z$ and q . Hence, we get a system of four polynomials with the other three polynomials in similar form as Equation 9 with the coefficients denoted by b_1 to b_{12}, c_1 to c_{12} , and d_1 to d_{12} .

We use the hidden variable resultant [2] method to solve for the unknowns in the system of polynomial. The idea is to group the unknown variable q with the known coefficients, i.e. ‘‘hidden variable’’ so that the system of polynomials can be rearrange into

$$\mathcal{Z}(q)\mathbf{X} = 0 \quad (10)$$

where $\mathbf{X} = [\hat{t}_x, \hat{t}_y, \hat{t}_z, 1]^T$ and $\mathcal{Z}(q)$ is a 4×4 matrix formed from the polynomial coefficients a, b, c, d and the unknown variable q . Since $\mathcal{Z}(q)$ is a square matrix, we know from linear algebra that Equation 10 has a non-trivial solution only if the determinant of $\mathcal{Z}(q)$ becomes zero, i.e.

$$\det(\mathcal{Z}(q)) = 0 \quad (11)$$

This determinant, which is otherwise known as the hidden variable resultant, gives an 8-degree univariate polynomial

$$Aq^8 + Bq^7 + Cq^6 + Dq^5 + Eq^4 + Fq^3 + Gq^2 + Hq + I = 0 \quad (12)$$

where $A, B, C, D, E, F, G, H, I$ are formed from the coefficients a, b, c, d of the system of polynomials. The univariate polynomial can be solved using the Companion matrix [2] or Sturm bracketing [12] methods which gives up to 8 real solutions. We chose the Companion matrix method in our implementation because it is easier to implement although it is less efficient than the Sturm bracketing method. The system of polynomials becomes linear once q is solved, and it becomes easy to solve for the remaining unknowns \hat{t}_x, \hat{t}_y and \hat{t}_z . Finally, we recover the rotation matrix \hat{R}_y with the known q from Equation 8.

4.3. Recover Relative Pose

It is important to note that the rotation matrix \hat{R}_y and translation vector \hat{t} solved in the previous section are not the relative pose of the two generalized camera frames V' and V . The relative pose $[R, t]$ has to be computed from $[\hat{t}, \hat{R}_y]$, and the roll and pitch angles which we have removed earlier. We compute the relative rotation R from Equation 3, and the relative translation t can be found from the expression

$[t]_x R_r'^T R_p'^T = R_r'^T R_p'^T [\hat{t}]_x$ which we have used earlier to factorize out $R_r'^T R_p'^T$ from the generalized essential matrix. This gives

$$t = R_r'^T R_p'^T \hat{t} \quad (13)$$

4.4. Degenerated Case

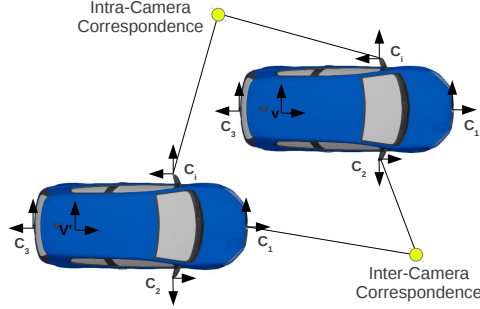


Figure 2. Illustration of intra- and inter-camera correspondences.

Our minimal 4-point algorithm degenerates when the multi-camera system undergoes pure translation while having only intra-camera correspondences. Figure 2 shows an illustration of the intra- and inter-camera correspondences. From the generalized epipolar constraint in Equations 1 and 2, and dropping the camera and image point indices for brevity, we have

$$u^T E u + (t'_C \times u')^T R u + u'^T R (t_C \times u) = 0 \quad (14)$$

We see that in the case of pure translation and intra-camera correspondences, i.e. $R = I$ and $t'_C = t_C$, $(t'_C \times u')^T R u + u'^T R (t_C \times u)$ becomes $(t_C \times u')^T u + u'^T (t_C \times u)$, which is zero because of the anti-symmetry of the triple product. Here, $t'_C \leftrightarrow t_C$ are the translation components of the extrinsics from the correspondent pair of cameras that see the line correspondence $l' \leftrightarrow l$. As a result, the epipolar constraint from Equation 14 becomes

$$u'^T E u = 0 \quad (15)$$

which is equivalent to the conventional epipolar constraint [5] where the scale can no longer be estimated.

We note that the univariate polynomial from Equation 12 gives $q = 0$ in the degenerated case. This means that we are able to identify the degenerated case with our minimal 4-point algorithm and obtain the relative translation without metric scale denoted by \hat{t} . We use an additional inter-camera correspondence, i.e. $t'_C \neq t_C$ to circumvent the scale problem in the degenerated case. Here, we solve for the unknown scale s from the following equation

$$s u'^T [\hat{t}]_x u + (t'_C \times u')^T u + u'^T (t_C \times u) = 0 \quad (16)$$

It is important to note that in practice under presence of noise, q takes a small value (not exactly 0) in the degenerated case. In our implementation, we take it as a degenerated case when $q \leq 0.01$. This value was obtained from simulation under the condition of 1 pixel noise.

5. Linear 8-Point Algorithm

In this section, we show that it is also possible to solve for \hat{t} and \hat{R}_y from Equation 6 linearly with eight Plücker line correspondences. Here we write

$$\hat{R}_y = \begin{bmatrix} \cos \hat{\theta} & -\sin \hat{\theta} & 0 \\ \sin \hat{\theta} & \cos \hat{\theta} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

Putting \hat{R}_y and \hat{t} into the generalized essential matrix \hat{E}_{GC} , we get

$$\hat{E}_{GC} = \begin{bmatrix} -e_1 & -e_2 & e_3 & r_1 & -r_2 & 0 \\ e_2 & -e_1 & -e_4 & r_2 & r_1 & 0 \\ e_5 & e_6 & 0 & 0 & 0 & r_3 \\ r_1 & -r_2 & 0 & 0 & 0 & 0 \\ r_2 & r_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_3 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

which consists of only nine unique entries, i.e.

$$\begin{aligned} e_1 &= \hat{t}_z \sin \hat{\theta}, & e_2 &= \hat{t}_z \cos \hat{\theta}, & e_3 &= \hat{t}_y, & e_4 &= \hat{t}_x, \\ e_5 &= \hat{t}_x \sin \hat{\theta} - \hat{t}_y \cos \hat{\theta}, & e_6 &= \hat{t}_x \cos \hat{\theta} + \hat{t}_y \sin \hat{\theta}, \\ r_1 &= \cos \hat{\theta}, & r_2 &= \sin \hat{\theta}, & r_3 &= 1 \end{aligned}$$

Putting the generalized essential matrix from Equation 18 back into the generalized epipolar constraint from Equation 2, we get

$$[g_1 \ g_2 \ g_3 \ g_4 \ g_5 \ g_6 \ g_7 \ g_8 \ g_9] \mathcal{E} = 0 \quad (19)$$

where g_1 to g_9 are the coefficients formed from the Plücker line correspondence and $\mathcal{E} = [e_1, e_2, e_3, e_4, e_5, e_6, r_1, r_2, r_3]^T$. With eight Plücker line correspondences, we get the following linear system

$$\mathcal{A} \mathcal{E} = 0 \quad (20)$$

where \mathcal{A} is a 8×9 matrix. An unique solution for Equation 20 exists when the rank of \mathcal{A} equals to 8. In this case, the solution is obtained via the standard SVD method [5] and enforcing the constraint of $r_3 = 1$, i.e. divide \mathcal{E} with its last element.

We note that in the case where there is no difference in the roll and pitch angles between the relative generalized camera frames, i.e. $R_r' = R_r$ and $R_p' = R_p$, and only intra-camera correspondences, the rank of \mathcal{A} drops to 7. As a result, the solution found from the standard SVD method is no longer unique but given by a family of solutions $(\lambda \hat{E}_{GC}, \lambda \hat{R}_y + \mu I)$, where the ambiguity lies only in the rotation matrix \hat{R}_y . We circumvent this problem by adopting the strategy proposed in [11] since similar problem has also been observed in the linear 17-point algorithm. Since the ambiguity in the solution lies only in the rotation matrix, the idea is to enforce the constraint of $\|e\| = 1$

on the elements of the essential matrix instead of $\|\mathcal{E}\| = 1$ while minimizing \mathcal{AE} using the SVD method. Formally, we are minimizing

$$\|\mathcal{A} [e^T, r^T]^T\| \quad s.t. \quad \|e\| = 1 \quad (21)$$

This is equivalent to solving

$$(\mathcal{A}_r \mathcal{A}_r^+ - I) \mathcal{A}_e e = 0 \quad (22)$$

where \mathcal{A}_e and \mathcal{A}_r are made up of the first six and last three columns of \mathcal{A} respectively, and \mathcal{A}_r^+ is the pseudo-inverse of \mathcal{A}_e . An unique solution for e can be obtained by applying the SVD method on the matrix $(\mathcal{A}_r \mathcal{A}_r^+ - I) \mathcal{A}_e$.

The conventional essential matrix \hat{E} is obtained from the first 3×3 matrix block in \hat{E}_{GC} with e . This means that the unique solution for \hat{R}_y and \hat{t} can be solved from the decomposition of \hat{E} and cheirality considerations [5]. Note that the solution for \hat{t} has no scale ambiguity. In the case of pure translation, \hat{t} is computed up to scale and one additional inter-camera point is needed to retrieve the metric scale using the method described in Section 4.4. Finally, the relative motion R and t can be recovered with the method described in Section 4.3.

6. Robust Estimation

Table 1. Comparisons on total number of iterations needed for RANSAC ($w = 0.5$ and $p = 0.99$) and evaluation of solutions.

Algorithm	RANSAC Loops	# of Solutions	Total
Minimal 4-Point	71	8	568
Linear 8-Point	1177	1	1177
Linear 17-Point	603606	1	603606
Minimal 6-Point	292	64	18688
*Ackermann 2-Point	17	6	102

*Does not give the full 6 degree-of-freedom for relative motion estimate.

We apply the RANSAC algorithm [3] for robust estimation to reject outlier correspondences as well as to determine the correct solution from the multiple solutions obtained from the minimal 4-point algorithm. The main factors that affects the computation cost, i.e. total number of iterations needed, are the number of correspondences needed for the problem and the number of solutions to be evaluated. The number of iterations k needed in RANSAC is given by $k = \frac{\ln(1-p)}{\ln(1-w^n)}$, where n is the number of correspondences needed for the algorithm (for example $n = 4$ for the 4-point algorithm), w is the probability that any selected correspondence is an inlier and p is the probability that all the selected correspondences are inlier. The total number of iterations m needed to run RANSAC while evaluating all the s number of solutions is given by $m = k \times s$. This is the worst case complexity. Assuming that $w = 0.5$ and $p = 0.99$, we need $71 \times 8 = 568$ iterations for our minimal 4-point algorithm and $1177 \times 1 = 1177$ iterations for our linear 8-point algorithm. We compare this with the linear 17-point

algorithm [13] which requires $603606 \times 1 = 603606$ iterations and minimal 6-point algorithm [15] which requires $292 \times 64 = 18688$ iterations. The high number of iterations needed for the linear 17-point and minimal 6-point algorithms made it impossible for practical applications. Although the Ackermann 2-point algorithm [9] requires only $17 \times 6 = 102$ iterations, it does not give the full 6 degree-of-freedom for the motion estimate (see Section 7 for more comparisons). Table 1 shows the summary of the comparison of the total number of iterations needed for the different methods.

7. Results

We show the accuracy of our proposed minimal 4-point and Linear 8-point algorithms by comparing its performances with other existing algorithms under various noisy conditions in simulations. We also show the feasibility of our minimal 4-point algorithm on real-world datasets.

7.1. Simulations

Pixel Noise: The accuracy of our proposed minimal 4-point and linear 8-point algorithms are compared with the existing linear 17-point [13] and Ackermann 2-point [9] algorithms. In addition, we do the comparison with the Gröbner basis solution for our minimal 4-point formulation generated from the automatic generator [7]. We make the simulated multi-camera system realistic by following the number of cameras, intrinsics and extrinsics parameters from our car setup in Figure 1. For each trial, the relative motion of the multi-camera system is randomly chosen from the range of $\theta = [0.1, 0.2]$ rad and $\rho = [0.5, 1.0]$ m, where θ and ρ are the relative yaw and scale following the Ackermann motion defined in [9]. The 3D points are also randomly generated within the range of $[-10, 10]$ m in the respective axis of the first multi-camera system frame. The randomly generated 3D points are reprojected onto the camera to get the image points. We make sure that each 3D point is seen by at least one camera over the two consecutive frames.

A total of 1000 trials each are carried out for the different image pixel noise with standard deviation ranging from 0.1 to 1.0 pixel at an interval of 0.1 pixel. Following the definition in [14], the relative translational error is defined as $2\|t - \tilde{t}\| / (\|t\| + \|\tilde{t}\|)$, where t and \tilde{t} are the estimated and ground truth translations. The relative rotational error is computed as the norm of the Euler angles from $R\tilde{R}^T$ where R and \tilde{R} are the estimated and ground truth rotation matrices. Figures 3(a) and 3(b) shows the average translational and rotational errors against the pixel noise. The linear 17-point algorithm shows the highest errors and the highest increment in error with the increment of the pixel noise. Our minimal 4-point and the Gröbner basis solution display very similar errors. This is because the Gröbner basis solution is

derived from the exact formulation of our minimal 4-point algorithm. Our linear 8-point and the Ackermann 2-point algorithms show the lowest errors, but it should be noted that the Ackermann 2-point algorithm has the disadvantage of not able to estimate the full 6 degree-of-freedom in the relative motion. All the minimal 4-point, Gröbner basis, linear 8-point and Ackermann 2-point algorithms show relatively little increment in errors with the increment of the pixel noise.

IMU Noise: We test the resilience of our minimal 4-point and linear 8-point algorithms against the presence of IMU noise in both the roll and pitch axes. The simulation setup is similar to the test on pixel noise except that now we fix the pixel noise standard deviation at 0.5 pixel and add noise to the roll and pitch from the IMU over the range of $[0, 0.6]$ deg. The upper bound of the noise is chosen to follow the noise from the lower grade MEMS IMUs such as the one used for quadrotors². The higher end IMUs such as the iTrace system used in our car system has an upper bound for roll and pitch noise at 0.01 deg.

Figures 3(c)-3(f) show the average errors from the 1000 trials at each IMU roll and pitch noise levels of our minimal 4-point and linear 8-point algorithms compared with the linear 17-point and Gröbner basis algorithms. The average errors from the linear 17-point algorithm remains stable over the different noise level. This is because the linear 17-point algorithm does not depend on the roll and pitch readings from the IMU. It can be seen from the plots that our minimal 4-point and linear 8-point algorithms do not deteriorate much with the increase in IMU noise and still stay within the error range of the linear 17-point algorithm. Similar to the pixel noise simulation, the Gröbner basis errors follow our minimal 4-point algorithm closely.

Off Ackermann Tests: We compare the performances of our minimal 4-point and linear 8-point algorithms with the Ackermann 2-point algorithm in the cases where the Ackermann motion constraint is violated. The simulation setup is similar to the test on pixel noise except that now we fix the pixel noise standard deviation at 0.5 and relative motion at $\theta = 0.1$ and $\rho = 0.5$. Three sets of tests are carried out by varying the (a) relative roll from $[0, 5]$ deg, (b) relative pitch from $[0, 5]$ deg and (c) relative z from $[0, 2]$ m. Figures 4(a)-4(f) show huge increase in the errors from the Ackermann 2-point algorithm with the off Ackermann motion in the relative roll, pitch and z axes respectively while the errors from our minimal 4-point and linear 8-point algorithms remain approximately constant. This shows the advantage of our minimal 4-point and linear 8-point algorithms over the Ackermann 2-point algorithm for motion estimation.

7.2. Real-World Datasets

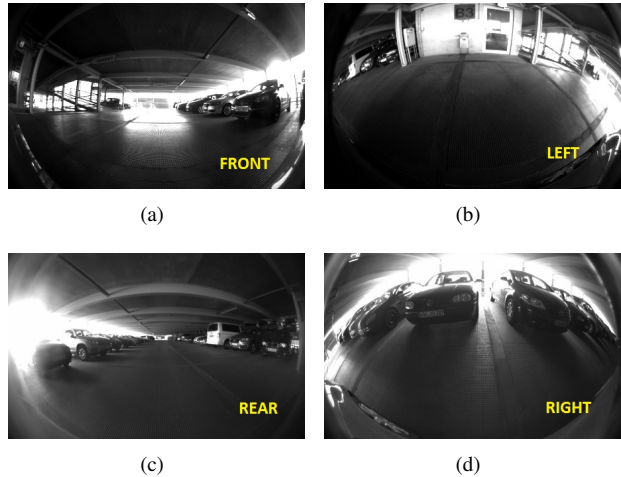


Figure 5. Sample images from the multi-camera system on our car.

We demonstrate the feasibility of our minimal 4-point algorithm by showing results from three real-world datasets collected from our car (see Figure 1) equipped with the multi-camera system and iTrace IMU for roll and pitch angles in an indoor multi-level parking garage. Figure 5 shows an example of the images collected from our multi-camera system. Note that we chose to collect the datasets while the car transits between different levels to highlight the strength of our minimal 4-point algorithm in estimating the full 6 degree-of-freedom of relative motion over the existing Ackermann 2-point algorithm. We extract the SURF features [1] and match them over consecutive frames. These feature correspondences are used to estimate the relative motion with our minimal 4-point algorithm. The relative motions are then concatenated together to form the full trajectory. Similar to [9], we also implemented four independent Kalman filters for t_x, t_y, t_z and θ respectively to smooth the estimated trajectory. Full bundle adjustment³ is also used to refine the trajectory and reconstructed 3D structures.

Figures 6(a), 7(a) and 8(a) show the full trajectories and reconstructed 3D structures of the scenes from the three datasets. We show the accuracy of the results with the reconstructed 3D structures due to the absence of ground truth trajectory. The first dataset was collected when the car moved up a ramp as shown in Figure 6(c), the second dataset was collected when the car moved down a ramp as shown in Figure 7(c) and the third dataset was collected when the car moved down two consecutive ramps shown in Figure 8(c). The datasets consist of a total of 309×4 , 298×4 and 498×4 images that span over approximately 50 m, 55 m and 85 m respectively. Figures 6(b), 7(b) and 8(b) show the transitions between different levels in the parking garage from the xz views of the trajectories.

²<http://www.asctec.de/>

³<http://code.google.com/p/ceres-solver/>

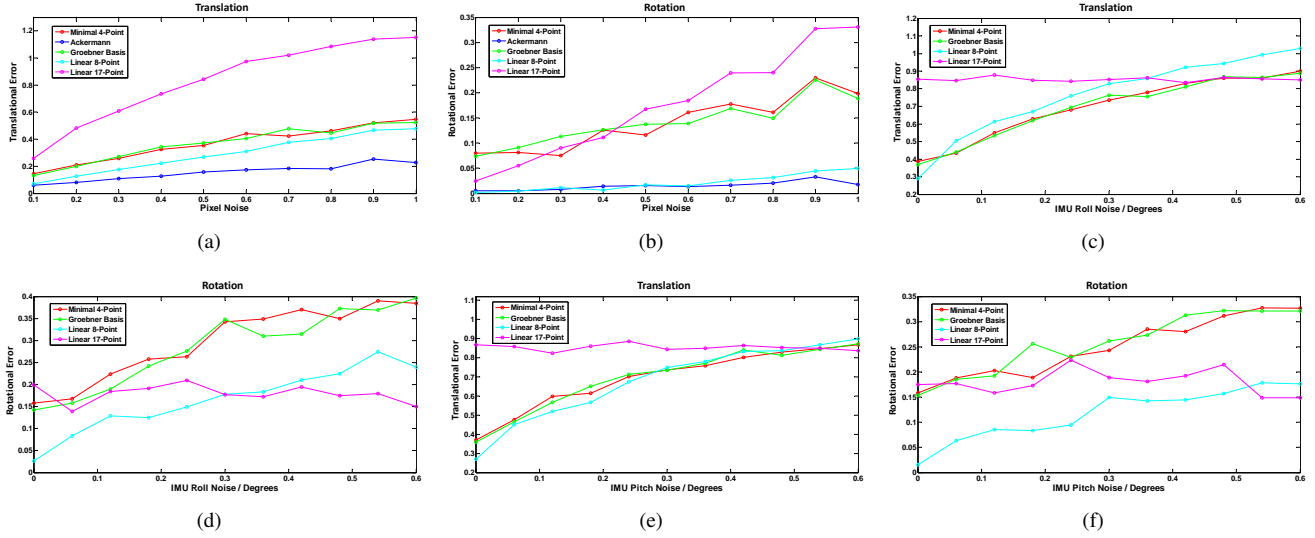


Figure 3. Comparison of the translational (no units) and rotational (rad) errors over (a)-(b) image noise in pixel, (c)-(d) IMU roll and (e)-(f) pitch angle noise in degrees with a fixed 0.5 pixel image noise.

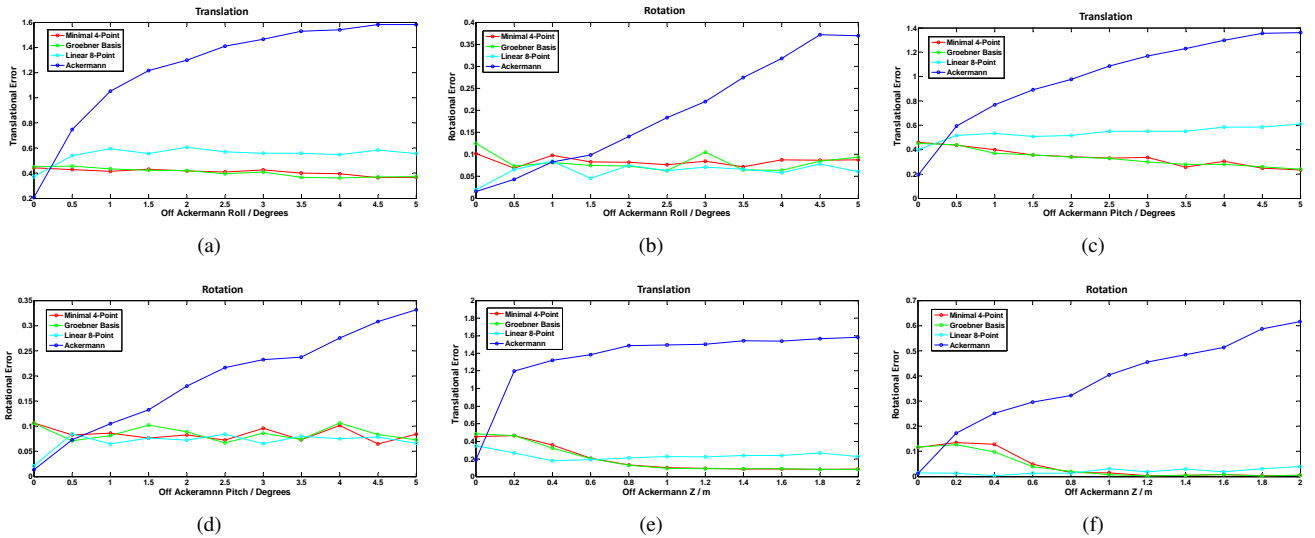


Figure 4. Comparison of the translational (no units) and rotational (rad) errors over off Ackermann motion in the relative (a)-(b) roll angles, (c)-(d) pitch angles and (e)-(f) z axis with a fixed 0.5 pixel image noise.

8. Conclusion

We have proposed two different algorithms to solve for the relative motion of a multi-camera system with known vertical, i.e. known absolute roll and pitch angles. In particular, we derived the minimal 4-point and linear 8-point algorithms. We show that in comparison to other existing methods, our proposed algorithms are practical to be used within RANSAC for real-world ego-motion estimation and free from any constraints, i.e. the full 6 degree-of-freedom for the relative motion is obtained. The results from both simulations and real-world datasets demonstrated the feasibility of our proposed algorithms.

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 *CVIU*, volume 110, pages 346–359, June 2008.
- [2] D. A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra*. Springer, 1997.

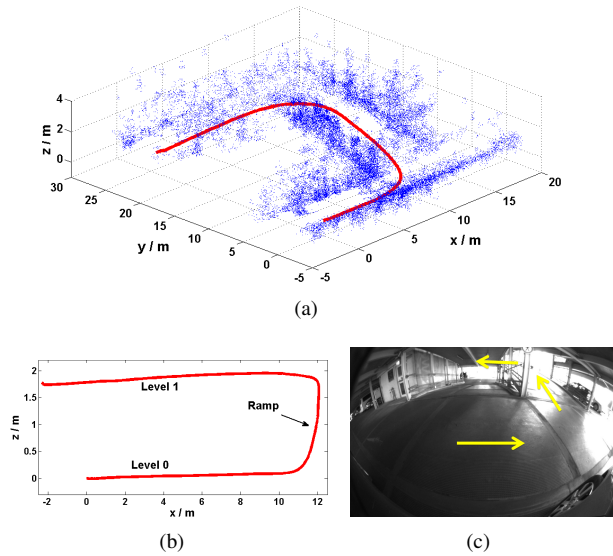


Figure 6. Car moving up a ramp. (a) Reconstructed 3D structures and trajectory starting at $(0, 0, 0)$. (b) xz view of the trajectory showing the difference in height. (c) Image of the upward ramp.

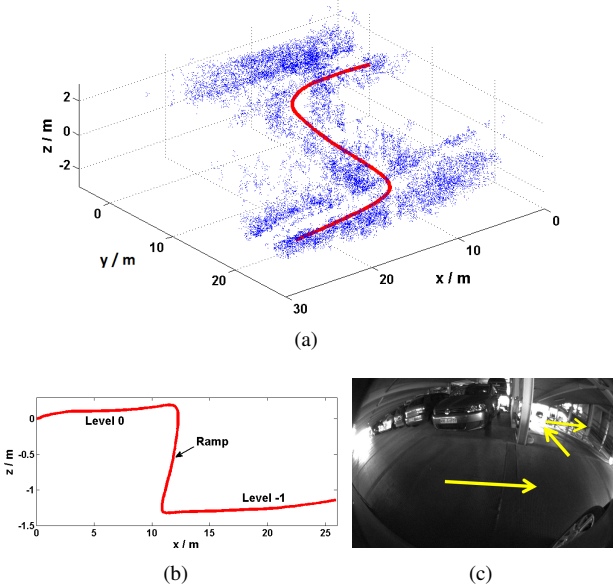


Figure 7. Car moving down a ramp. (a) Reconstructed 3D structures and trajectory starting at $(0, 0, 0)$. (b) xz view of the trajectory showing the difference in height. (c) Image of the downward ramp.

[3] 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.

[4] F. Fraundorfer, P. Tanskanen, and M. Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *ECCV*, pages 269–282, 2010.

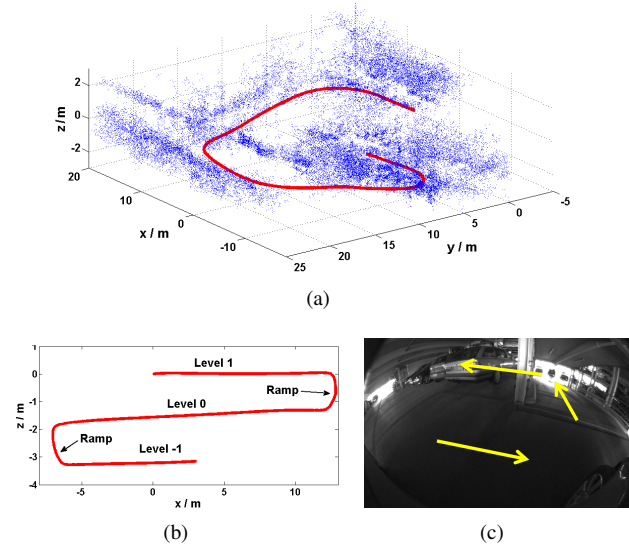


Figure 8. Car moving down two ramps. (a) Reconstructed 3D structures and trajectory starting at $(0, 0, 0)$. (b) xz view of the trajectory showing the difference in heights. (c) Image of the first ramp.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2nd edition, 2004.

[6] M. Kalantari, A. Hashemi, F. Jung, and J.-P. Guedon. A new solution to the relative orientation problem using only 3 points and the vertical direction. In *Journal of Mathematical Imaging and Vision*, volume 39, pages 259–268, March 2011.

[7] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *ECCV*, pages 302–315, 2008.

[8] Z. Kukelova, M. Bujnak, and T. Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In *ACCV*, 2010.

[9] G. H. Lee, F. Fraundorfer, and M. Pollefeys. Motion estimation for a self-driving car with a generalized camera. In *IEEE CVPR*, June 2013.

[10] G. H. Lee, F. Fraundorfer, and M. Pollefeys. Structureless pose-graph loop-closure with a multi-camera system on a self-driving car. In *IEEE/RSJ IROS*, 2013.

[11] H. Li, R. Hartley, and J. Kim. A linear approach to motion estimation using generalized camera models. In *IEEE CVPR*, pages 1–8, June 2008.

[12] D. Nistér. An efficient solution to the five-point relative pose problem. In *PAMI*, volume 26, pages 756–777, June 2004.

[13] R. Pless. Using many cameras as one. In *IEEE CVPR*, volume 2, pages 587–93, June 2003.

[14] L. Quan and Z. D. Lan. Linear n-point camera pose determination. In *PAMI*, volume 21, pages 774–780, 1999.

[15] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *OMNIVIS*, 2005.

[16] P. Sturm. Multi-view geometry for general camera models. In *IEEE CVPR*, volume 1, pages 206–212, June 2005.