Motion Estimation for Self-Driving Cars With a Generalized Camera

Gim Hee Lee[1], Friedrich Fraundorfer[2], Marc Pollefeys[1]

Department of Computer Science[1]
ETH Zürich, Switzerland
{glee@student, pomarc@inf}.ethz.ch

Faculty of Civil Engineering and Surveying[2]
Technische Universität München, Germany
friedrich.fraundorfer@tum.de

Abstract

In this paper, we present a visual ego-motion estimation algorithm for a self-driving car equipped with a close-to-market multi-camera system. By modeling the multi-camera system as a generalized camera and applying the non-holonomic motion constraint of a car, we show that this leads to a novel 2-point minimal solution for the generalized essential matrix where the full relative motion including metric scale can be obtained. We provide the analytical solutions for the general case with at least one inter-camera correspondence and a special case with only intra-camera correspondences. We show that up to a maximum of 6 solutions exist for both cases. We identify the existence of degeneracy when the car undergoes straight motion in the special case with only intra-camera correspondences where the scale becomes unobservable and provide a practical alternative solution. Our formulation can be efficiently implemented within RANSAC for robust estimation. We verify the validity of our assumptions on the motion model by comparing our results on a large real-world dataset collected by a car equipped with 4 cameras with minimal overlapping field-of-views against the GPS/INS ground truth.

1. Introduction

Self-driving cars such as those featured in the DARPA Urban Challenge [4], rely heavily on sensors like Radar, Lidar and GPS to perform ego-motion estimation, localization, mapping and obstacle detection. In contrast, cameras are only playing a minor role in self-driving cars but are already commonly found in commercially-off-the-shelf (COTS) cars for driving and/or parking assistance. An example is the Nissan Quasquai Around View Monitor where four cameras are mounted on the car to provide full omni-directional view around the car. Although such camera system is currently used only for driving and/or parking assistance, it offers huge potential to be the main sensor for self-driving cars without the need for major modifications.

Motivated by the fact that multi-camera systems are already available in some COTS cars, we focus this paper on using a multi-camera setup for ego-motion estimation which is one of the key features for self-driving cars. A multi-camera setup can be modeled as a generalized camera as described by Pless [12]. In this work, he derived the generalized epipolar constraint (GEC) and it was shown in [8] that the full relative motion can be obtained with metric scale. We make use of the fact that the generalized camera that is rigidly fixed onto a car that has to adhere by the Ackermann steering principle [15] to simplify the GEC and design an efficient and robust algorithm for visual ego-motion estimation.

Figure 1. Car with a multi-camera system consisting of 4 cameras (front, rear and side cameras in the mirrors).

We show that two point correspondences are sufficient to estimate the generalized essential matrix with metric scale by using the Ackermann motion model (i.e. circular motion on a plane) where there are 2 free parameters - scale and yaw angle for the relative motion between 2 consecutive frames. Consequently, we derive the analytical 2-point minimal solution for the general case with at least one inter-camera correspondence and a special case with only intra-camera correspondences. A maximum of up to 6 solutions exists for the relative motion in both cases. The small number of necessary point correspondences and solutions makes it suitable for robust estimation with RANSAC and real-time operations. We show that the scale can always be recovered from the general case with at least one inter-

camera correspondence and identify the existence of degeneracy when the car undergoes straight motion in the special case with only intra-camera correspondences where the scale cannot be determined. We use the special case with only intra-camera correspondences as the default case for our implementation since there is always more intra-camera correspondences than inter-camera correspondences. We propose a practical method to retrieve the scale when the car undergoes straight motion from one additional inter-camera correspondence and the known yaw angle which essentially reverts the special case to the general case.

The relative motions can be concatenated together to get the full trajectory of the car. We implement Kalman filters with constant velocity prior to smooth out noisy estimates for real-time operations. Finally, we relax the Ackermann motion constraint by doing a full 6 degrees of freedom (DOF) pose-graph [11] optimization for loop-closure followed by bundle adjustment for all the poses and 3D points. We verify our approach by comparing our results on a large real-world dataset collected from a generalized camera setup that consists of 4 cameras mounted on a car (see Figure 1) looking front, rear, left and right with minimal field-of-views against the GPS/INS ground truth.

Our main contributions can be summarized as follows:

- A practical ego-motion estimation algorithm with metric scale from a new formulation of the generalized essential matrix using the GEC and Ackermann motion model.
- Analytical 2-point minimal solutions for the general case with at least one inter-camera correspondence and a special case with only intra-camera correspondences.
- An investigation and a practical solution to the degenerate case of straight motion with only intra-camera correspondences.

2. Related Works

Our work builds on top of previous works about generalized cameras and it is also related to other works with multi-camera systems that are not using the generalized camera formulation.

The idea of a generalized camera system where a single epipolar constraint is used to describe the relative motion of a set of cameras mounted rigidly on a single body over two different frames was first proposed by Pless [12]. The main difference between a generalized camera system and a single pinhole camera is the absence of a single center of projection. Pless derived a generalized essential matrix, which is a 6x6 matrix with 18 unknowns from the GEC. He suggested a linear 17-point algorithm to solve for the generalized essential matrix. Sturm also showed similar results in [17]. However, both works did not show any results from real-world data.

Li $et\ al.$ [8] extended the work on the GEC by identifying the degenerated cases for the locally-central generalized camera setup. The locally-central generalized camera refers to a configuration where the cameras only do intra-camera correspondences. In contrast to the general case where the generalized camera does inter-camera correspondences, Li $et\ al.$ showed that the rank of the GEC drops from 17 to 16 because the null motion always satisfies the GEC. He noted that this solution is often found using the standard Singular Value Decomposition (SVD) method [6] to solve the GEC linearly. He suggested a new linear approach to solve the GEC despite the degeneracy. He also pointed out that the same approach can be used to solve for the GECs in the degenerated cases of axial and locally-central-and-axial-cameras. He showed results from a small-scale dataset collected with a Point-Grey ladybug camera in a controlled laboratory environment. The proposed linear algorithms need 17 or 16 point correspondences, a number, that induced high computational cost when creating a robust estimator using RANSAC.

A minimal solution for the generalized essential matrix, suitable for RANSAC hypothesis generation, was proposed by Stewénius $et\ al.$ [16]. The derived minimal solution uses 6-point correspondences to solve the GEC problem. The method involves solving a polynomial equation system and results in 64 solutions. The high number of solutions also puts high computational costs on a robust estimator like RANSAC. Nevertheless, a RANSAC implementation of the 6-point minimal solution was shown on synthetic datasets but not on any real-world dataset.

In comparison with the works from Pless, Li $et\ al.$ and Stewénius $et\ al.$, we proposed the 2-point algorithm from the combination of the GEC and Ackermann motion model. And for the first time, this allows an efficient motion estimation of a generalized camera system with a robust estimator like RANSAC on a large real-world dataset.

The motion model used has previously been used by Scaramuzza $et\ al.$ [13] where they proposed the 1-point RANSAC algorithm for conventional monocular visual odometry on a car. Similarly, they made used of the Ackermann motion model in the epipolar constraint and this reduced the number of free parameters to two. However, they used one omnidirectional camera with a single center of projection and this prohibited the retrieval of metric scale. The validity of the 1-point algorithm also came with the constraint that the camera must be placed along the back-wheel axis of the car. In an extension [14], Scaramuzza $et\ al.$ developed a method where the motion and metric scale could be computed with a 2-point algorithm from a single monocular camera. The camera has to be placed with an offset to the back-wheel axis and this offset needs to be known. Similar to our algorithm for generalized camera with only intra-camera correspondences, the scale becomes

unobservable for straight motion. In contrast, we propose a practical method to retrieve the metric scale when the car is moving straight with our formulation using the generalized camera setup.

Other methods such as [2, 7] estimated the relative motions of the multi-camera setups without using the GEC. In [2], Clipp *et al.* estimated the relative motion without the scale using the 5-point algorithm [10] in one camera, while the metric scale is retrieved from an additional point from another camera. He showed results from both simulations and a real-world dataset. However, a major limitation is that the scale can only be retrieved under certain conditions. In [7], Kazik *et al.* estimated the relative motions of two cameras with non-overlapping field of view by using the 5-point algorithm [10]. The metric scale is retrieved by using the 'hand-eye' calibration constraints. The success of this method relies heavily on the estimation from the 5-point algorithm in the individual cameras. They showed results with small-scale datasets collected in controlled laboratory environments. No discussion on how their method could be extended to more than two cameras was provided.

3. Generalized Camera Model

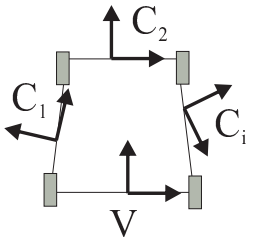


Figure 2. Illustration of a generalized camera on a car.

In this section, we briefly describe the concept of the generalized camera model which is essential for understanding the remaining paper. More details can be found in [12, 17]. Figure 2 shows an illustration of a generalized camera setup on a car. It is made up of individual cameras denoted by C_i that are mounted rigidly on arbitrary locations on the car. The generalized camera has a reference frame denoted by V. Let us denote the intrinsics and extrinsics of the respective cameras by K_i and $T_{C_i} = [R_{C_i}\ t_{C_i};\ 0\ 1]$. The normalized image coordinate of a point $\mathbf{x}_{ij}$ is then given by $\hat{\mathbf{x}}_{ij} = K_i^{-1}\mathbf{x}_{ij}$. The dependency on a single camera projection center to describe a 3D point X_j is removed by using the 6-vector Plücker line

$$\mathbf{l}_{ij} = [\mathbf{u}_{ij}^T \quad (t_{C_i} \times \mathbf{u}_{ij})^T]^T \tag{1}$$

which describes the light ray that connects $\mathbf{x}_{ij}$ and X_j. $\mathbf{u}_{ij} = R_{C_i}\hat{\mathbf{x}}_j$ is the unit direction of the ray expressed in the reference frame V. This changes the point correspondences from 2 image coordinates to 2 intersecting rays and, as shown in [12], the epipolar constraint now becomes

$$\mathbf{l}_{ij,k+1}^T \underbrace{\begin{bmatrix} E & R \\ R & 0 \end{bmatrix}}_{E_{GC}} \mathbf{l}_{ij,k} = 0 \tag{2}$$

where $\mathbf{l}_{ij,k}$ and $\mathbf{l}_{ij,k+1}$ are the correspondence Plücker lines from frame k and $k+1$. E_{GC} is the generalized essential matrix from the GEC. R is the rotation matrix between the generalized camera reference frames at k and $k+1$. E follows the conventional essential matrix [6] decomposition $E = \lfloor t \rfloor_{\mathbf{x}} R$ where t is the translation vector between frame k and $k+1$. It is important to note that t is determined only up to scale from the conventional essential matrix for a single camera but the metric scale can be fully determined using the generalized camera as shown in [8].

4. Motion Estimation

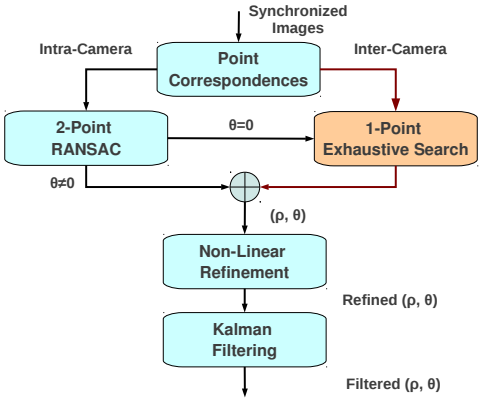


Figure 3. System overview for motion estimation of the generalized camera on a car.

Figure 3 shows the overview of the pipeline for the estimation of the relative motion between two consecutive car frames using the generalized camera. A set of synchronized images is obtained from the generalized camera. Intra-camera point correspondences are computed from these images. Our 2-Point RANSAC algorithm always computes the relative motion based on the intra-camera correspondences. In the case where the relative yaw angle θ is found to be near zero from the 2-Point RANSAC, the inter-camera point correspondences are extracted and used to compute the scale ρ from a 1-point exhaustive search. Note that θ is kept fixed in the 1-point search. The estimated ρ and θ are further refined using the non-linear refinement and Kalman filtering steps.

4.1. Point Correspondences

Our motion estimation algorithm relies on two sets of point correspondences - intra-camera and inter-camera. Specifically, intra-camera point correspondences refer to correspondences which are seen by the same camera over

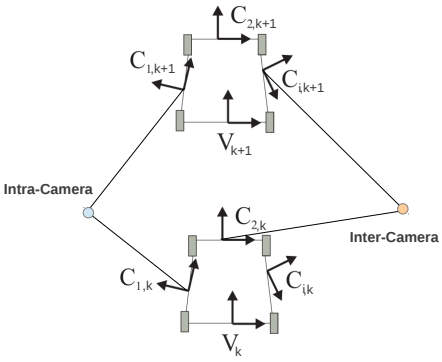


Figure 4. Example of intra- and inter-camera point correspondences.

two consecutive frames and inter-camera refers to correspondences which are seen by different cameras over two consecutive frames as illustrated in Figure 4. We extract and match SURF [1] features on the GPU for both intra-camera and inter-camera. In principle, our generalized camera configuration on the car always allows inter-camera point correspondence. This is because part of the scene from the front camera will be seen by the left and right cameras at the next frame. Similarly, part of the scenes from the left and right cameras will always be seen by the rear camera in the next frame. The number of inter-camera correspondences are however far lesser than intra-camera correspondences. Hence, we chose to use intra-camera correspondences as the default case for our implementation and rely on one-additional inter-camera correspondences to retrieve the scale in the degenerated case when the car is moving straight. In the very rare occasion where no inter-camera inliers are found, we propagate the scale from the previous estimate with the Kalman filter (see Section 4.6).

4.2. 2-Point Minimal Solution

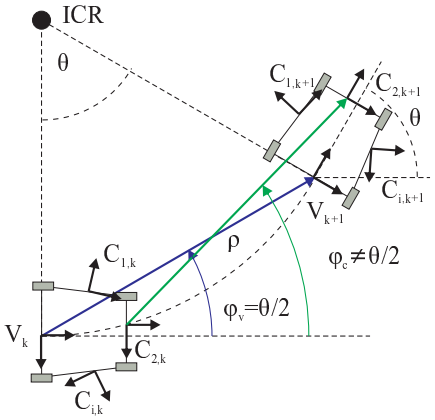


Figure 5. Relation between generalized camera in Ackermann motion.

Figure 5 shows the illustration of a generalized camera on a car that undergoes the Ackermann motion over 2 consecutive frames k and $k+1$. Specifically, the car undergoes a circular motion about the Instantaneous Center of Rotation (ICR) with the Ackermann model. The radius of the circular motion goes to infinity when the car is moving straight. The main objective of motion estimation is to compute the relative motion R and t between V_k and V_{k+1}. Following the derivation in [13], using V_k as the reference frame, it can be observed from the diagram that

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad t = \rho \begin{bmatrix} \cos\varphi_v \\ \sin\varphi_v \\ 0 \end{bmatrix} \tag{3}$$

where θ is the relative yaw angle and ρ is the scale of the relative translation. Here, the z-axis of V_k is pointing out of the paper. It can be further observed from the diagram that φ_v is the angle between ρ and the perpendicular line to the radius of the circle at V_k, hence $\varphi_v = \frac{\theta}{2}$. We immediately see that the relative motion between frame V_k and V_{k+1} is dependent on only 2 parameters - scale ρ and yaw angle θ.

Putting Equation 3 into E_{GC} from Equation 2, we get

$$E_{GC} = \begin{bmatrix} 0 & 0 & \rho\sin\frac{\theta}{2} & \cos\theta & -\sin\theta & 0 \\ 0 & 0 & -\rho\cos\frac{\theta}{2} & \sin\theta & \cos\theta & 0 \\ \rho\sin\frac{\theta}{2} & \rho\cos\frac{\theta}{2} & 0 & 0 & 0 & 1 \\ \cos\theta & -\sin\theta & 0 & 0 & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

For brevity, let us now drop all the indices on the Plücker line vector from Equation 2 and simply denote 2 correspondence Plücker lines as $\mathbf{l} = [\mathbf{u}^T \ \ (t_C \times \mathbf{u})^T]^T$ from frame k and $\mathbf{l}' = [\mathbf{u}'^T \ \ (t_{C'} \times \mathbf{u}')^T]^T$ from frame $k+1$. Equation 2 can then be written as

$$a\cos\theta + b\sin\theta + c\rho\cos\frac{\theta}{2} + d\rho\sin\frac{\theta}{2} + e = 0 \tag{5}$$

where

$$\begin{aligned} a &= -u_w(t_{Cx}u'_y - t_{Cy}u'_x) - u'_w(t_{C'x}u_y - t_{C'y}u_x) \\ &\quad + u_y(t_{C'w}u'_x - t_{Cw}u'_x) + u_x(t_{Cw}u'_y - t_{C'w}u'_y) \\ b &= u_x(t_{C'x}u'_w - t_{C'w}u'_x) + u_y(t_{C'y}u'_w - t_{C'w}u'_y) \\ &\quad - u_x(t_{Cx}u_w - t_{Cw}u_x) - u'_y(t_{Cy}u_w - t_{Cw}u_y) \\ c &= u_w u'_y - u_y u'_w \\ d &= u_x u'_w + u_w u'_x \\ e &= u_w(t_{C'x}u'_y - t_{C'y}u'_x) + u'_w(t_{Cx}u_y - t_{Cy}u_x) \end{aligned}$$

Here, the subscripts x, y and w refer to the components in the vector. Equation 5 is our new GEC with the Ackermann motion model. We need 2 Plücker line correspondences to solve for the 2 unknowns ρ and θ in Equation 5. Denoting each set of known coefficients obtained from each Plücker line correspondence by $(a_1, b_1, c_1, d_1, e_1)$ and $(a_2, b_2, c_2, d_2, e_2)$, and using the trigonometric half-angle formula

$$\cos\theta = 1 - 2\sin^2\frac{\theta}{2} \tag{6a}$$

$$\sin\theta = 2\sin\frac{\theta}{2}\cos\frac{\theta}{2} \tag{6b}$$

we get

$$\rho = \frac{-e_1 - a_1(1-2\beta^2) - b_1(2\alpha\beta)}{c_1\alpha + d_1\beta} \tag{7a}$$

$$\rho = \frac{-e_2 - a_2(1-2\beta^2) - b_2(2\alpha\beta)}{c_2\alpha + d_2\beta} \tag{7b}$$

where $\alpha = \cos\frac{\theta}{2}$ and $\beta = \sin\frac{\theta}{2}$. Combining Equations 7a and 7b to eliminate ρ, we get

$$\begin{aligned}(2a_1\beta^2 - 2b_1\alpha\beta - e_1 - a_1)(c_2\alpha + d_2\beta) - \\ (2a_2\beta^2 - 2b_2\alpha\beta - e_2 - a_2)(c_1\alpha + d_1\beta) = 0\end{aligned} \tag{8}$$

where the Pythagorean identity from Equation 9 has to be satisfied.

$$\sin^2\frac{\theta}{2} + \cos^2\frac{\theta}{2} = \alpha^2 + \beta^2 = 1 \tag{9}$$

Using the Sylvester Resultant [3] method to eliminate $\alpha = \cos\frac{\theta}{2}$ from the two polynomials in Equations 8 and 9, we get a 6 degrees polynomial equation in terms of $\beta = \sin\frac{\theta}{2}$ which can be further reduced to a cubic polynomial by making $\gamma = \beta^2$. The roots of the cubic polynomial is obtained in closed-form by using the cubic formula.

$$A\beta^6 + B\beta^4 + C\beta^2 + D = 0 \tag{10a}$$

$$A\gamma^3 + B\gamma^2 + C\gamma + D = 0 \tag{10b}$$

A, B, C and D are known coefficients made up of $(a_1, b_1, c_1, d_1, e_1)$ and $(a_2, b_2, c_2, d_2, e_2)$. We drop the full expressions of A, B and C for brevity but show D because it has a special property.

$$\begin{aligned}D = -c_2^2(e_1^2 + a_1^2) - 2c_2^2e_1a_1 - c_1^2(e_2^2 + a_2^2) - 2c_1^2e_2a_2 \\ +2c_2c_1(a_1a_2 + e_1e_2) + 2c_2c_1(e_1a_2 + a_1e_2)\end{aligned} \tag{11}$$

An interesting observation is that when we do purely intra-camera correspondences, i.e. $t_c = t_{c'}$, the last 2 terms of coefficient a cancel out and $a = -e$. Putting this new relation into Equation 11, we see that all terms cancel out and $D = 0$. Hence, Equation 10b becomes

$$\gamma(A\gamma^2 + B\gamma + C) = 0 \tag{12}$$

where one of the solution for γ is always 0 and the remaining two solutions from the quadratic polynomial are given by $\gamma = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$. Putting γ back into the relation $\gamma = \beta^2$, we get up to a maximum of six real solutions for β where two are always 0. Finally, the yaw angle θ of the relative motion can be computed from $\beta = \sin\frac{\theta}{2}$ and the scale ρ can be be computed from Equation 7. We have verified that both Equation 10 and 12 are the minimal-degree polynomials by computing the Gröbner Basis [3].

It was mentioned earlier that it is easier to get many more reliable correspondences with intra- than inter-camera. In addition, it is more efficient to compute the roots from the quadratic polynomial from Equation 12 than the cubic polynomial from Equation 10. Therefore, we adopted the intra-camera point correspondences strategy in our implementation.

4.3. Degenerated Case: Metric Scale Computation

The metric scale of the relative motion cannot be uniquely determined from the GEC when the car is moving straight i.e. $\theta = 0$ with only intra-camera correspondences. This can be observed by substituting $\theta = 0$ into Equation 7 where the numerator cancels out since $a = -e$. This means that ρ is always 0 hence cannot be uniquely determined for $\theta = 0$. Nonetheless, we can still uniquely identify that $\theta = 0$ by assigning unit scale i.e. $\rho = 1$ for the solution of $\theta = 0$. This is because an unit scale still fulfills the Sampson error [6] computation within RANSAC (see next section). The correct solution yields the highest number of inliers from RANSAC.

It is important to note that the scale can always be uniquely determined from the GEC when there is at least one inter-camera point correspondence. We can be easily see this by putting $t_c \neq t_{c'}$ into the coefficients a and e from Equation 5 where we observed that $a \neq -e$. Hence, ρ can be uniquely determined from Equation 7. This suggests that we can make use of one additional inter-camera point correspondence to find the metric scale when $(\rho = 1, \theta = 0)$ turns out to be the solution with the highest inliers from the pure intra-camera correspondences case. In practice, this can be done effectively by doing an exhaustive search through all inter-camera point correspondences for inliers.

4.4. Robust Estimation

We make our 2-point algorithm robust by implementing it within RANSAC [5] to effectively reject outliers. We do this by checking the Sampson error [6] for each point correspondence within the individual camera. The essential matrix of each individual camera can be computed from the hypotheses of the relative motion R and t between the car reference frame V and the extrinsics T_{Ci} of the camera. The number of iterations m needed in RANSAC is given by $m = \frac{\ln(1-p)}{\ln(1-\upsilon^n)}$ where n is the number of correspondences needed to form the hypothesis, p is the probability that all selected features are inliers and υ is the probability that any selected correspondence is an inlier. Assuming that $p = 0.99$ and $\upsilon = 0.5$, a total of 16 iterations are needed for our 2-point algorithm. We compare this with the 6-, 16- and

17-point algorithms which need 292, 301802 and 603606 iterations respectively. The total number of iterations needed for our algorithm, including the additional 1-point exhaustive search, is still far lower than the number needed by the 6-, 16- and 17-point algorithms.

4.5. Non-Linear Refinement

A non-linear refinement is applied using all the inliers that were found from RANSAC to get a better estimate of ρ and θ. The cost function for the non-linear refinement over two consecutive frames k and $k+1$ is given by

$$\underset{X,\rho,\theta}{\operatorname{argmin}} \sum_{i,i' \in \mathcal{C}} \sum_{j} \{||\pi(P_i, X_j) - \mathbf{x}_{ij}||^2 + ||\pi(P_{i'}, X_j) - \mathbf{x}_{i'j}||^2\} \tag{13}$$

where $(\mathbf{x}_{ij} \leftrightarrow \mathbf{x}_{i'j})$ are the point correspondences from camera C_i and $C_{i'}$ over frame k and $k+1$. X_j is the triangulated 3D point from $(\mathbf{x}_{ij} \leftrightarrow \mathbf{x}_{i'j})$. The set $\mathcal{C}$ gives the intra- and inter-camera indices for all the point correspondences over frame k and $k+1$. $\pi(.)$ is the projection function that projects the 3D point onto the image. P_i and $P_{i'}$ are the camera projection matrices given by

$$P_i = K_i[R_{C_i}^T \quad -R_{C_i}^T t_{C_i}] \tag{14a}$$

$$P_{i'} = K_{i'}[R_{C_{i'}}^T R^T \quad -R_{C_{i'}}^T (R^T t + t_{C_{i'}})] \tag{14b}$$

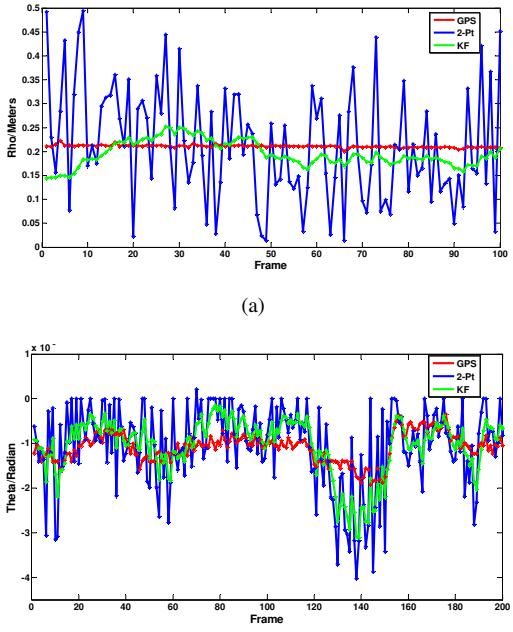
K, R_C and t_C are the intrinsics and extrinsics of the camera. R and t are the relative motion of the car defined by Equation 3 which are functions of the parameters ρ and θ we are optimizing over. The initial values for non-linear refinement are taken from the RANSAC hypothesis and its triangulated 3D points.

4.6. Kalman Filtering

We implement the Kalman filter with constant velocity prior for both the scale ρ and yaw angle θ to smooth out any noisy estimation from our algorithm. We know that the control inputs for a car involves independent steering and linear speed. This means that two independent 1D Kalman filters can be applied to smooth out the estimates for the scale ρ and yaw angle θ respectively. Figure 6 shows examples of the relative motions of our car from the 2-point algorithm (blue line) and the smoothed estimates from the Kalman filters (green line). We compare both estimates with the GPS/INS ground truth (red line) and it can be seen that the outputs from the Kalman filter follow more closely to the GPS/INS ground truth.

5. Results

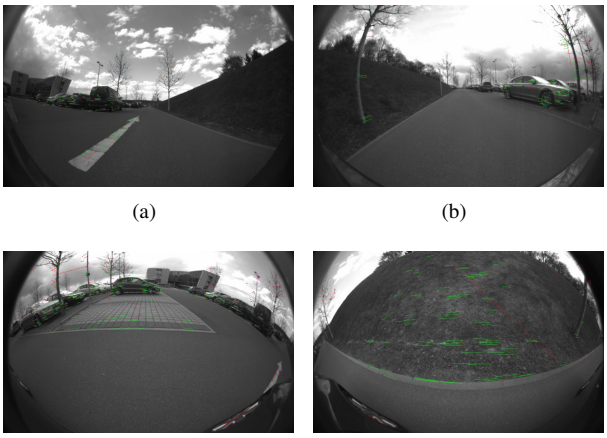
Figure 1 shows a picture of the car used to collect the dataset for testing our algorithm. Four cameras with fish-eye lens are mounted onto the car on the front and rear of



(a)

(b)

Figure 6. Example of scales ρ (a) and yaw angles θ (b) between consecutive frames after Kalman filtering.



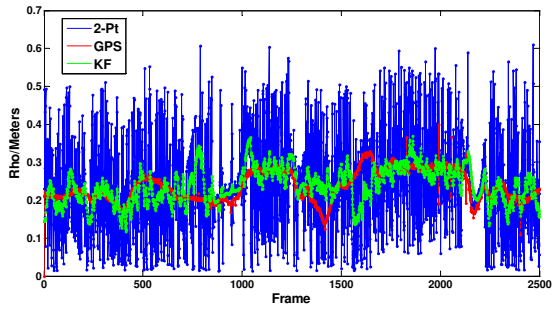
(a) (b)

(c) (d)

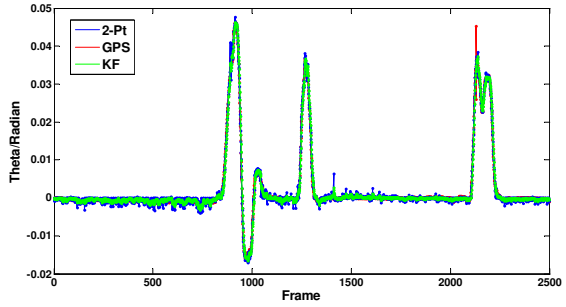
Figure 7. Images from the 4 cameras with fish-eye lens on the car. (a) Front, (b) Rear, (c) Left, (d) Right.

the chassis, and the two sides in the mirror holders. The intrinsics of the cameras are calibrated with [9] and the extrinsics are provided by the car manufacturer. The dataset was collected while driving the car in a loop that is about 600m around a car park. The dataset consists of mostly static objects with a few moving pedestrians. GPS/INS readings of the trajectory were also captured during the drive for ground truth. A total of 4×2500 images are used in the test of our algorithm. Figure 7 shows an example of the images from all the cameras for a frame. The inlier (green line) and outlier (red line) point correspondences from our 2-point algorithm are also shown on the images. The SURF keypoints and descriptors are extracted and matched over the

raw fish-eye images. These keypoints are undistorted using the fish-eye camera model from [9] before they are used for triangulation to get the 3D points.



(a)



(b)

Figure 8. Scales ρ (a) and yaw angles θ (b) between all consecutive frames from our motion estimation algorithm compare with GPS/INS ground truth.

Figure 8(a) and 8(b) are the plots of all the 2499 relative motions - scales and yaw angles estimated with our 2-point algorithm. The blue lines are the estimated values from our 2-point algorithm, the green lines are the estimated values from after Kalman filtering and the red lines are the GPS/INS ground truth. Figure 8(b) shows that the car is moving straight for most of the trajectory except for three major turns at around frame 900, 1300 and 2200. These are the segments of the trajectory where both the scales and yaw angles are estimated solely with intra-camera point correspondences. An additional inter-camera correspondence is used to compute the scale for 79.9% of the trajectory when the car is moving straight. Figure 8(a) shows that the scale estimations are very close to the GPS/INS ground truth even without Kalman filtering. The remaining straight or almost straight relative motions are degenerated or near-degenerated cases where the yaw angles are computed from the 2-point algorithm with the intra-camera point correspondences and the scales are computed from the 1-point exhaustive search with the inter-camera feature correspondences. Figure 8(a) shows that the estimated scales from the 1-point exhaustive search are noisier with a standard deviation of around 0.125m from GPS/INS ground truth. Here, the Kalman filter helps to smooth out the noise. Figure 8(b) shows that the estimates for the yaw angles follow closely to the GPS/INS ground truth even without Kalman filtering.

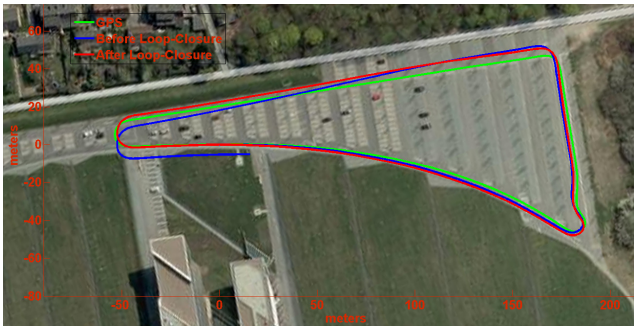


Figure 9. Trajectories before and after pose-graph loop-closure compared with GPS/INS ground truth.

The relative motions estimated with our algorithm are concatenated together to form the full trajectory of the car. The blue line on Figure 9 shows the trajectory recovered from the relative motions estimated with Kalman filtering. The accumulated drifts resulted in a loop-closure error which we removed by performing the pose-graph loop-closure [11]. The trajectory after loop-closure (red line) is significantly closer to the GPS/INS ground truth (green line). Finally, we do a full bundle adjustment over the whole trajectory and all the reconstructed 3D points. Note that we relaxed the Ackermann constraint and do the full 6 DOF optimization over the car poses in the loop-closure and bundle adjustment. Figure 10 shows the top view of the final trajectory and 3D points after bundle adjustment. The pose-graph loop-closure and bundle adjustment are implemented with the Google Ceres Solver [1].

We implemented the full pipeline on a Intel Core2 Quad CPU @ 2.40GHz $\times$ 4 with 4G of memory and GeForce GTX 285 GPU. The runtime is 6 fps not including pose-graph loop-closure and full bundle adjustment. Images from the cameras come at 12 fps and this means that a real-time implementation on the car is possible if we skip every other frame.

6. Conclusion

In this paper, we demonstrated visual ego-motion estimation for a car equipped with a multi-camera system with minimal field-of-views. The camera system was modeled as a generalized camera and we showed that the generalized essential matrix simplifies significantly when constraining the motion to the Ackerman motion model (i.e. circular motion on a plane). We derived an analytical 2-point minimal solution for the general case with at least one inter-camera correspondence and a special case with only intra-camera correspondences. We showed that a maximum of up to 6 solutions exists for the relative motion in both cases. We investigated the degenerate case of straight motion with

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

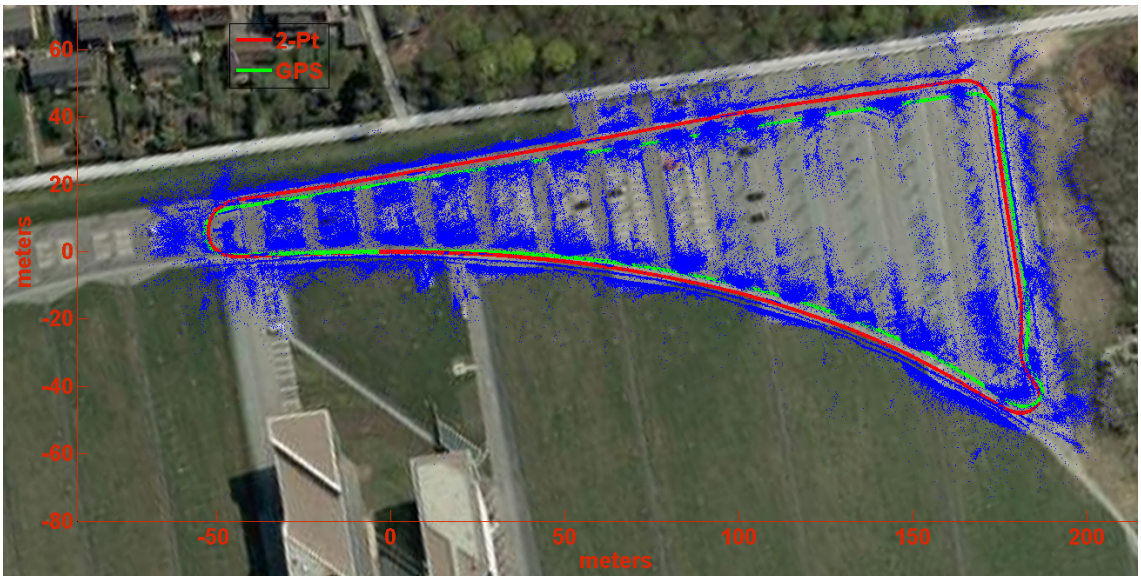


Figure 10. Top view of trajectory and 3D map points after pose-graph loop-closure and full bundle adjustment compared with GPS/INS ground truth.

intra-camera correspondences (which appears frequently in real data) and presented a practical solution using one additional inter-camera feature correspondence. We evaluated our method on a large real-world dataset and compared it to GPS/INS ground truth. The results of the comparison clearly showed that our assumptions on the vehicle motion hold for real-world data.

7. 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] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley. Robust 6dof motion estimation for non-overlapping, multi-camera systems. In *Workshop on the Applications of Computer Vision*, pages 1–8, January 2008.

[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] Defense Advanced Research Projects Agency. DARPA Urban Challenge 2007. `http://archive.darpa.mil/grandchallenge/`, November 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. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[7] T. Kazik, L. Kneip, J. Nikolic, M. Pollefeys, and R. Siegwart. Real-time 6d stereo visual odometry with non-overlapping fields of view. In *Computer Vision and Pattern Recognition*, pages 1529–1536, June 2012.

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

[9] C. Mei and P. Rives. Calibrage non biaise d'un capteur central catadioptrique. In *RFIA*, January 2006.

[10] D. Nistér. An efficient solution to the five-point relative pose problem. In *Pattern Analysis and Machine Intelligence*, volume 26, pages 756–777, June 2004.

[11] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *International Conference on Robotics and Automation*, pages 2262–2269, 2006.

[12] R. Pless. Using many cameras as one. In *Computer Vision and Pattern Recognition*, volume 2, pages 587–93, June 2003.

[13] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *International Conference on Robotics and Automation*, pages 4293 –4299, May 2009.

[14] D. Scaramuzza, F. Fraundorfer, R. Siegwart, and M. Pollefeys. Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In *International Conference on Computer Vision*, pages 1–7, September 2009.

[15] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots*. MIT Press, 2nd edition, 2011.

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

[17] P. Sturm. Multi-view geometry for general camera models. In *Computer Vision and Pattern Recognition*, volume 1, pages 206–212, June 2005.