Bundle adjustment using single-track vehicle model

Bundle Adjustment using Single-Track Vehicle Model

Jonas Nilsson[1,2] , Jonas Fredriksson[2] and Anders C.E. Ödblom[1]

Abstract— **This paper describes a method for estimating the 6-DoF viewing parameters of a calibrated vehicle-mounted camera. Visual features are combined with standard in-vehicle sensors and a single-track vehicle motion model in a bundle adjustment framework to produce a jointly optimal viewing parameter estimate. Results show that the vehicle motion model in combination with in-vehicle sensors exhibit good accuracy in estimating planar vehicle motion. This property is preserved, when combining these information sources with vision. Furthermore, the accuracy obtained from vision-only in direction estimation is not only maintained, but in fact further improved, primarily in situations where the matched visual features are few.**

I. Introduction

Robust and accurate methods for estimating the pose, i.e. position and direction, of a moving vehicle have many applications, for instance navigation or map building. Another application is *augmented imagery*, which is generated by adding virtual objects into real images. Augmented imagery is useful for many purposes, e.g. motion pictures, sensor evaluation, [1], and augmented reality, [2]. When creating realistic augmented imagery, the pose of the camera needs to be estimated with high precision in order to place the virtual objects accurately in the image.

Using an image sensor for pose estimation and structure reconstruction, so called Structure from Motion (SfM), [3], is a cost-efficient approach when creating augmented imagery since it requires no additional hardware. The most common approach for single camera SfM is to match image features between consecutive image frames and, based on a geometric camera model, estimate the camera poses and 3D positions of the observed features. The main disadvantage with SfM is that it fails when image information is poor, i.e. when image features cannot be matched robustly between consecutive frames. To improve accuracy, an estimate can be optimized numerically using *Bundle Adjustment* (BA). BA improves accuracy and robustness, as shown in e.g. [4] and [5], but requires a sufficiently accurate initial estimate to converge.

We propose to combine the high availability of vehicle motion models and standard in-vehicle sensor data, i.e. wheel speeds, yaw rate and steering wheel angle, with the accuracy of visual BA to create a robust and accurate pose estimate for a vehicle-mounted camera. The main scientific contribution is the inclusion of a single track vehicle motion model in a bundle adjustment framework.

The remainder of this paper is organized as follows. First, related work is presented followed by descriptions of the camera and vehicle models. Section IV describes how the models are used in a bundle adjustment framework, including parametrization and error models. In Section V the complete algorithm for pose estimation is briefly described. Finally, Section VI presents the results from experiments with recorded data and Section VII states the conclusions.

II. RELATED WORK

Traditionally, SfM is performed without real-time constraints and therefore processing the image sequence as a batch is a feasible and often preferred approach. Batch methods, e.g. bundle adjustment, are accurate but often computationally demanding. Consequently, recursive techniques are better suited for real-time applications. Simultaneous Localization and Map building (SLAM) is a commonly used term for pose estimation and map building in robotic applications, see e.g. [6] for an overview. SLAM has a history of adopting diverse sensor types and also various motion models, and a majority of the approaches have used recursive filtering techniques, such as the Extended Kalman Filter (EKF). The term Visual Odometry (VO), [7], refers to performing pose estimation using image sensors only.

Recursive techniques are computationally efficient but, in contrast to e.g. BA, do not consider the complete history of measurements concurrently, making them less accurate. Thus, real-time SLAM techniques have been complemented by local BA, e.g. [8], where the most recent pose estimates are refined by visual features. BA can also be extended to simultaneously not only optimize visual features, but also other sensor observations, as done for instance in [9] with inertial sensors and in [10] with GPS.

The remainder of this section will focus on ground vehicle-based SfM/Visual SLAM/VO approaches. For ground vehicle pose estimation, stereo approaches are clearly dominant in literature, see e.g. [5], [7]. A lot of work is also focused on single omnidirectional cameras, e.g. [11], [12]. In [12] an Ackerman steering model is used to parameterize the motion between two consecutive frames using a single point correspondence. For instance it is shown how using a motion model increase robustness to moving objects in the image.

Camera pose estimation using single standard cameras, i.e. cameras with medium to small field-of-view, is more

This work was supported by the Swedish Automotive Research Program FFI - Vehicle and Traffic Safety, and the Vehicle and Traffic Safety Centre (SAFER)

[1]J. Nilsson and A.C.E. Ödblom are with the Vehicle Dynamics and Active Safety Centre, Volvo Car Corporation, 40531 Göteborg, Sweden jnilss94@volvocars.com, aodblom1@volvocars.com

[2]J.Nilsson and J. Fredriksson are with the Department of Signals and Systems, Chalmers University of Technology, 41296 Göteborg, Sweden jonas.fredriksson@chalmers.se

challenging, compared to omnidirectional and stereo approaches. Single standard cameras are common in many applications due to reasons of e.g. cost or system complexity. In [8] a vision-only approach is demonstrated which employ sliding window BA. Validation is performed in an urban environment with an average speed of around 20 km/h.

III. Modeling

This section presents models used for vehicle pose estimation. The first subsection introduces notation and describes the used reference frames. These reference frames are needed to link the local camera and vehicle models, presented in the remaining subsections, to the global vehicle pose, which we seek to estimate.

A. Coordinate Systems

There are four different reference frames used in this paper, see Figure 1. Denote by G the fixed global frame and by C the moving reference frame which has its origin in the camera center of projection. Furthermore, let V denote the moving reference frame centered in the vehicle Center of Gravity (CoG) and aligned with the vehicle. Note that the moving reference frames C and V are time dependent, meaning that e.g. the velocity of the vehicle CoG in the reference frame V is equal to zero by definition. Describing vehicle dynamics is preferably done in a fixed reference frame. For this purpose, we denote by i a fixed reference frame, aligned with the vehicle, at time t_i. The reference frames V and i will thus coincide at time t_i.

The relation between the coordinates for a point in space can be described by the Euclidian transformation

$$P_A = R_{AB} P_B + C_{BA}, \tag{1}$$

where $P_A = \begin{bmatrix} x_A & y_A & z_A \end{bmatrix}^T$ and $P_B = \begin{bmatrix} x_B & y_B & z_B \end{bmatrix}^T$ are the point coordinates in the Cartesian reference frames A and B respectively, R_{AB} is a rotation matrix, and C_{BA} is a vector describing pure translation.

Rotations are described using three different parametrizations, see [13] for details. Besides rotation matrices, R, and unit quaternions, q, local Euler angles, in the form of roll, φ, pitch, θ, and yaw, ψ, angles are used, primarily to describe local vehicle direction.

B. Pinhole Camera

Consider the pinhole camera model, [3], where the image plane is located in $z_C = f$ and a point in space P_C is projected in $U = \begin{bmatrix} u & v \end{bmatrix}^T$ in the image plane. In the camera reference frame, the relation between a point in space and its corresponding projection in the image plane can be described by

$$\lambda U^h = K \begin{bmatrix} I & | & 0 \end{bmatrix} P_C^h, \ \lambda \in \mathbb{R}, \tag{2}$$

where $P_C^h = \begin{bmatrix} x_C & y_C & z_C & 1 \end{bmatrix}^T$ and $U^h = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ are the points expressed in homogenous coordinates and the matrix K is referred to as the *camera calibration matrix*.

While (2) is a linear equation in homogenous coordinates it becomes nonlinear in Cartesian coordinates. Denote the rows of the right-hand side of (2) as $\begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}^T$. Using this notation, Canceling λ from (2) gives

$$U = \frac{1}{r_3} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \tag{3}$$

which describe the projection of P_C in image coordinates.

In the case where the camera is vehicle-mounted, (2) can be expressed using the global vehicle pose and the relative pose between the camera and the vehicle as

$$\lambda U^h = K R_{CV} R_{VG} \begin{bmatrix} I & | & -C_{VG} - R_{GV} C_{CV} \end{bmatrix} P_G^h. \tag{4}$$

C. Local Vehicle Motion

This section describes the models for local vehicle motion. The lateral dynamics are described by the *bicycle* or *single-track* model, see [14] for a detailed description. Consider the lateral velocity $\dot{y}$ and the yaw rate $\dot{\psi}$ of the vehicle CoG. These states are defined according to Figure 2. Note that the reference frame is considered fixed at each time instant.

Let $\xi = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\psi} \end{bmatrix}^T$ be the vehicle dynamics state vector. The system dynamics are described by

$$\dot{\xi} = f_{dyn}(\xi, \delta) \tag{5}$$

where

$$f_{dyn}(\xi, \delta) = \begin{bmatrix} 0 \\ \frac{2}{m}\left(-\frac{m\dot{x}\dot{\psi}}{2} - \frac{\left(C_f + C_r\right)\dot{y} + \left(C_f l_f - C_r l_r\right)\dot{\psi}}{\dot{x}} + C_f \delta \right) \\ \frac{2}{I_z}\left(-\frac{\left(C_f l_f - C_r l_r\right)\dot{y} + \left(C_f l_f^2 + C_r l_r^2\right)\dot{\psi}}{\dot{x}} + C_f l_f \delta \right) \end{bmatrix}. \tag{6}$$

The front wheel steer angle, δ, and the distances l_f and l_r are defined according to Figure 2. The vehicle mass and yaw inertia are denoted m and I_z while C_f and C_r are the tire cornering stiffnesses for the front and rear wheel respectively.

Denote by ξ_i the vehicle dynamics state vector in the fixed vehicle reference frame i. The vehicle 3D position,

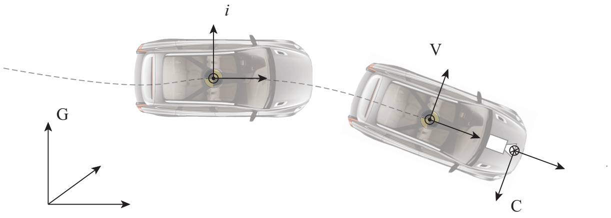

Fig. 1. Reference frames

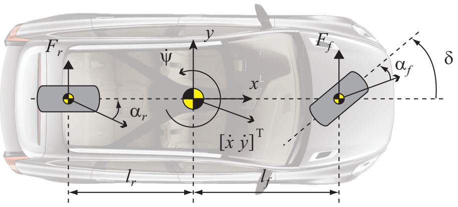

Fig. 2. Single-track vehicle model

roll, pitch and yaw is referred to as the vehicle pose state $\zeta_i = \begin{bmatrix} x_i & y_i & z_i & \varphi_i & \theta_i & \psi_i \end{bmatrix}^T$ and is described by

$$\dot{\zeta}_i = f_{pos}(\xi_i) = \begin{bmatrix} \dot{x}_i & \dot{y}_i & 0 & 0 & 0 & \dot{\psi}_i \end{bmatrix}^T. \tag{7}$$

To use the local vehicle motion in global pose estimation, the vehicle motion in the fixed local vehicle reference frame i, needs to be expressed in terms of the global vehicle pose. The local vehicle motion at time t is defined by the motion of the vehicle reference frame V in the fixed reference frame i. The local vehicle CoG pose, $\{R_{Vi}, C_{Vi}\}$, in the fixed vehicle frame i can be expressed as

$$R_{Vi} = R_{VG} R_{Gi} \tag{8a}$$

$$C_{Vi} = R_{iG} \left(C_{VG} - C_{iG} \right). \tag{8b}$$

Differentiating gives the corresponding derivatives

$$\dot{R}_{Vi} = \dot{R}_{VG} R_{Gi} \tag{9a}$$

$$\dot{C}_{Vi} = R_{iG} \dot{C}_{VG}, \tag{9b}$$

which describe the translational and angular velocities of the vehicle CoG in the fixed vehicle frame i.

IV. BUNDLE ADJUSTMENT

This section describes the BA framework and how image features, vehicle model, and in-vehicle sensors are incorporated in that framework.

BA is essentially a parameter estimation problem where the parameters β describe the camera motion and the position of elements in the surrounding environment. Consider observations acquired from error functions on the form

$$e_k(\beta) = \bar{g}_k - g_k(\beta). \tag{10}$$

Typically, $\bar{g}_k$ is a measurement or a mathematical relation and $g_k(\beta)$ is a prediction of $\bar{g}_k$ given the parameter vector β. The parameter estimation problem in BA is commonly solved using Maximum Likelihood (ML). Given the independent observations $e = \begin{bmatrix} e_1 & e_2 & \dots & e_n \end{bmatrix}$ the ML-estimate, $\hat{\beta}_{ML}$, is found by maximizing the Likelihood function $p(e|\beta)$. This is equivalent to minimizing the sum of negative log-Likelihood functions:

$$\hat{\beta}_{ML} = \underset{\beta}{\operatorname{argmin}} \sum_{l=1}^{n} - \ln p_l(e_l|\beta). \tag{11}$$

The following subsections define the parameter vector β and formulate the error functions e_k, $k \in \{vis, dyn, pos, \delta, \psi, v\}$, and their respective distributions.

A. Parametrization

The purpose of the estimation process is to, at each time instant t_i, estimate the global position, C_{iG}, and the inverse rotation quaternion, q_{iG}, of the vehicle CoG. To enable the use of the vehicle model from Section III-C in the optimization, additional vehicle states are added to the parameter vector. For each time instant a parameter vector is formulated as $\beta_i = \begin{bmatrix} C_{iG} & q_{iG} & \dot{x}_i(t_i) & \dot{y}_i(t_i) & \dot{\psi}_i(t_i) & \delta(t_i) \end{bmatrix}^T$, where $\dot{x}_i$, $\dot{y}_i$, $\dot{\psi}_i$ and δ are expressed in fixed vehicle coordinates and defined according to Section III-C.

The complete parameter vector $\beta = \begin{bmatrix} \beta_{map} & \beta_1 & \beta_2 & \dots & \beta_n \end{bmatrix}^T$, consists of parameters for each time instant and parameters describing 3D structure points where $\beta_{map} = \begin{bmatrix} P^h_{G,1} & P^h_{G,2} & \dots & P^h_{G,m} \end{bmatrix}^T$ and $P^h_{G,j}$ is a 3D structure point in global homogeneous coordinates.

B. Error Modeling

This section describes the error functions and their probability distributions. Each distribution is characterized by its covariance Matrix Σ. A summary of the selected distributions is found in Table I.

1) Image Feature Reprojection: Let $\bar{g}_{vis}$ be an image feature observation of the 3D structure point $P^h_{G,j}$, observed from a vehicle-mounted camera. The *reprojection* of $P^h_{G,j}$ in the image, $g_{vis}(\beta)$, is given by (3), (4) and Section IV-A. The reprojection error is defined as

$$e_{vis}(\beta) = \bar{g}_{vis} - g_{vis}(\beta). \tag{12}$$

To describe feature reprojection errors, the commonly used Gaussian distribution ($\mathcal{N}$) is a poor choice since it decays very quickly when moving far away from the mean, i.e. it has small tails. Typically, a significant part of the reprojection errors originate from incorrectly matched features, i.e. outliers, which often give rise to quite large reprojection errors. Here, the robust Gaussian distribution ($\mathcal{N}_R$), described in [15], is used. The Gaussian distribution is complemented by an underlying uniform distribution where the latter account for the presence of outliers.

2) Vehicle Motion Model: The relation between β, defined by Section IV-A, and the two state vectors ξ_i and ζ_i, defined in Section III-C, is given by (8) and (9). ξ at two time instants are connected by integration of (5),

$$\bar{g}_{dyn} = \xi_{i+1}(t_{i+1}) - \xi_i(t_i) - \int_{t_i}^{t_{i+1}} f_{dyn}(\xi(t), \delta(t)) dt = 0. \tag{13}$$

Denoting the integral above I_{dyn}, and approximating it using the trapezoidal rule yields

$$\begin{aligned} I_{dyn} &\approx I_{dyn}(\beta_i, \beta_{i+1}) = \\ &= \frac{h_i}{2} \left(f_{dyn}(\xi_i(\beta_i), \delta(t_i)) + f_{dyn}(\xi_{i+1}(\beta_{i+1}), \delta(t_{i+1})) \right), \end{aligned} \tag{14}$$

where $h_i = t_{i+1} - t_i$. The prediction as a function of β becomes

$$g_{dyn}(\beta) = \xi_{i+1}(\beta_{i+1}) - \xi_i(\beta_i) - I_{dyn}(\beta_i, \beta_{i+1}), \tag{15}$$

and the corresponding error function

$$e_{dyn}(\beta) = 0 - g_{dyn}(\beta). \tag{16}$$

The derivation of $e_{pos}(\beta)$ is analogous to that $e_{dyn}(\beta)$ shown above with the exception that all states, since ζ describe pose, must be expressed in the same reference frame.

The selected distributions $p(e_{dyn}|\beta)$ and $p(e_{pos}|\beta)$, shown in Table I, are based on the estimated approximation error in (14). This error will have contributions from modeling errors, non-linearities in the integrand and errors in the measured

TABLE I

ERROR PROBABILITY DISTRIBUTIONS. ALL DISTRIBUTIONS HAVE ZERO MEAN. IDENTITY MATRIX WITH SIZE $k \times k$ IS DENOTED $I_{k\times k}$

Variable	Distribution	$\Sigma^{1/2}$
e_{vis}	$\mathscr{N}_R$	$I_{2\times2}$
e_{dyn}	$\mathscr{N}$	$I_{3\times3}\sigma_{dyn}^T$
e_{pos}	$\mathscr{N}$	$\begin{bmatrix} I_{3\times3}\sigma_P^T & 0 \\ 0 & I_{3\times3}\sigma_D^T \end{bmatrix}$
e_v	$\mathscr{N}$	$0.01\bar{v}_i$
$e_{\dot\psi}$	$\mathscr{N}$	$0.1\frac{\pi}{180}$
e_δ	$\mathscr{N}$	$0.2\frac{\pi}{180}$
Variable	Value	
σ_{dyn}	$\begin{bmatrix} 10h_i & h_i & h_i \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}\sigma_h$	
σ_P	$\begin{bmatrix} 20\frac{h_i^3}{12} & 20\frac{h_i^3}{12} & 0.1h_i \end{bmatrix} + \begin{bmatrix} \bar{v}_i & 1 & 1 \end{bmatrix}\sigma_h$	
σ_D	$\begin{bmatrix} 2h_i & \infty & 140\frac{h_i^3}{12} \end{bmatrix}\frac{\pi}{180} + \begin{bmatrix} 5\frac{\pi}{180} & 5\frac{\pi}{180} & \bar{\dot\psi}_i \end{bmatrix}\sigma_h$	
σ_h	10^{-3}	

time difference h_i. In cases where the state derivative is also a state, there are no modelling errors. Therefore, the errors arising from a non-linear integrand are more dominant.

3) In-Vehicle Sensors: It is straightforward to predict in-vehicle sensor measurements by describing what parameters they measure. Front Steer angle and yaw rate are included in the parameter vector, see Section IV-A, and thus the error functions become

$$e_\delta(\beta_i) = \bar{\delta}_i - \delta(t_i) \tag{17}$$

and

$$e_{\dot\psi}(\beta_i) = \bar{\dot\psi}_i - \dot\psi_i(t_i). \tag{18}$$

The vehicle speed is available from wheel speed sensors and is modeled as a measurement of the planar vehicle velocity

$$e_v(\beta_i) = \bar{v}_i - \sqrt{\dot{x}_i(t_i)^2 + \dot{y}_i(t_i)^2}. \tag{19}$$

The chosen distributions in Table I are based on measurement data.

V. VISUAL RECONSTRUCTION ALGORITHM

This section briefly describes the algorithm, summarized by Algorithm 1, for estimating the vehicle pose.

From each image a set of image features are extracted using the Scale Invariant Feature Transform (SIFT), [16]. There exists a wide range of methods for estimating the relative pose between two image frames. Here, RANSAC in combination with the 8-point algorithm for calibrated cameras is used, [3]. At each RANSAC iteration, eight randomly chosen feature pairs are used to estimate the relative pose. The candidate solution with the greatest number of inliers, i.e. features with reprojection error < 2 pixels, is chosen as top candidate β_i.

Vehicle information is then used to improve the quality of the relative pose estimate. First, each pose candidate is scaled using wheel speed sensors. Second, a pose candidate is generated by a motion model simulation. Given the vehicle state at frame $i-1$, the vehicle state at frame i is obtained by simulating the system in (5)-(7) from t_{i-1} to t_i. Also, in-vehicle sensors are used in this simulation. This planar prediction is a useful complement when the visual reconstruction is inaccurate. Third, the error models used for bundle adjustment, extensively described in Section IV-B, are used to filter out candidates which are highly unlikely. Candidates with errors exceeding four standard deviations, excluding feature reprojection errors, are removed.

Algorithm 1 Visual reconstruction

```
initialize $\beta_1$
Extract features from frame 1
**for** $i = 2, 3, \ldots, n_{frames}$ **do**
    Extract features from frame $i$
    Match features between frames $i-1$ and $i$
    Estimate $\beta_i$
    Add new points to $\beta_{map}$
    Bundle adjustment of $\begin{bmatrix} \beta_{map} & \beta_{i-w+1} & \cdots & \beta_i \end{bmatrix}^T$
    Remove outliers from $\beta_{map}$
**end for**
```

Bundle Adjustment is performed on a frame window of size $w = 10$. The BA problem is solved numerically by using the Levenberg-Marquardt algorithm, see [15]. All feature correspondences originating from the frame window are initially included in the BA, thereby relying on the robust Gaussian distribution to ignore outliers.

VI. RESULTS

In this section, four different estimation algorithms are evaluated and compared experimentally.

A. *Sensors+Model*; simulation of planar vehicle motion using in-vehicle sensors and vehicle motion model
B. *Vision*; error models of visual features only
C. *Vision+Sensors*; error models of visual features and in-vehicle sensors
D. *Vision+Sensors+Model*; error models of visual features, in-vehicle sensors and vehicle motion model

The first algorithm differs from the others as it does not perform bundle adjustment and also has only 3-DoF, i.e. planar motion. It is included to show the performance of the vehicle motion model complemented with in-vehicle sensors. The latter three all perform bundle adjustment and are differentiated by the choice of error models. For details on error models, see Section IV-B. For these algorithms, the used method for initial estimation of each vehicle pose is identical, with the exception that the error models used for discarding unfeasible candidates are different, see Section V for details on this method.

A. Data Collection

Data has been collected using a Volvo V70 passenger vehicle with a calibrated forward facing monochrome camera with 640×480 pixel resolution, frame rate of $7.5 - 15$ frames/s and a horizontal field-of-view of 37°. All vehicle parameters are known and ground truth has been recorded from a system which fuse differential-GPS and Inertial Measurement Unit (IMU) data.

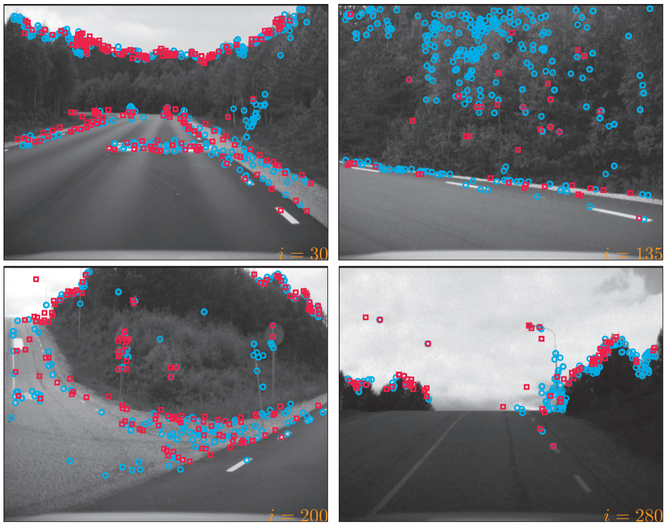

Fig. 3. Image examples with frame indices 30, 135, 200 and 280. Blue circles indicate detected features while red squares mark features which have been matched to corresponding features in at least one adjacent frame.

The 301-frame long test sequence contains very challenging situations from a vision perspective, where on some occasions no distant features are visible in the images. Also, large direction changes between consecutive frames, in situations such as sharp 90° turns, elevation changes and speed bumps, makes feature matching difficult. Speed bumps and elevation changes is also challenging for the vehicle motion model, as it assumes planar motion. Image examples are shown in Figure 3. Note particularly the top right frame, illustrating an image frame with few matched features and no distant features present. Figure 4 shows the number of matched visual features for all frames in the sequence.

B. Estimation Results

Performance is evaluated in different reference frames by comparing pose estimates to ground truth. Estimates are evaluated either in the fixed global reference frame G, as a trajectory, or in the local vehicle reference frame i. To compare trajectories, true and estimated states are assumed equal at the initial time instant. To compare local estimation errors, consider the local vehicle pose at time at t_{i+1}, expressed in the local vehicle reference frame at time t_i. Comparing this local vehicle pose estimate to ground truth yields a frame-by-frame error which display the performance in each local state.

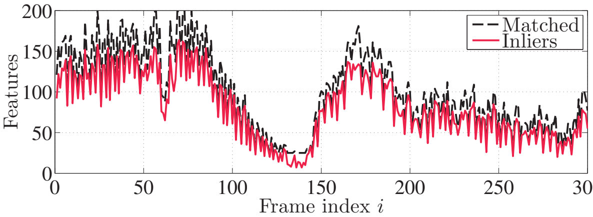

Fig. 4. Number of matched and inlier features. A feature is classified as matched if it has been matched to a feature in at least one adjacent frame. Feature observations with reprojection error < 1 pixel, using Algorithm D, are defined as inliers.

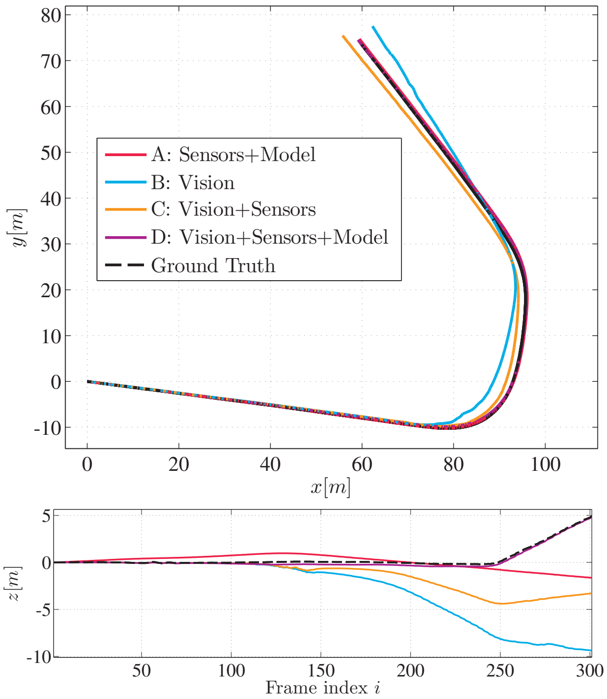

Fig. 5. Trajectory and altitude comparison.

Analyzing global vehicle trajectories shows the accumulated effect of errors over time. Local estimation errors will accumulate and affect the future trajectory estimate in a non-linear manner. This has the effect of greater influence from local estimation errors generated at the beginning of the trajectory compared to similar errors generated at the end of the trajectory.

Figure 5 visualizes the estimated trajectories from the four algorithms and show how algorithm A estimates the planar trajectory very well, while having no possibility to estimate altitude correctly. Note that the altitude estimates from algorithms $B-C$ start to diverge from ground truth around frame index 120, where algorithm B seems to diverge slightly more than algorithm C. This part of the sequence is challenging from a vision perspective, as indicated by the low number of matched features seen in Figure 4. Algorithm D, which use all available sources of information, clearly has the best altitude estimation.

Since local estimation errors all contribute non-linearly to the trajectory errors, analyzing what effects are causing these errors is difficult by studying the figures above. Therefore, local direction errors are presented in Figure 6 where the estimation difficulties around frame index 120 can be directly observed. In this part of the sequence, there are large errors which are gradually reduced by adding in-vehicle sensors and the motion model to the visual algorithm, i.e. algorithms $B-D$.

Specifically, adding in-vehicle sensors primarily improves the yaw estimate, $\hat{\psi}$, since yaw rate is measured, while

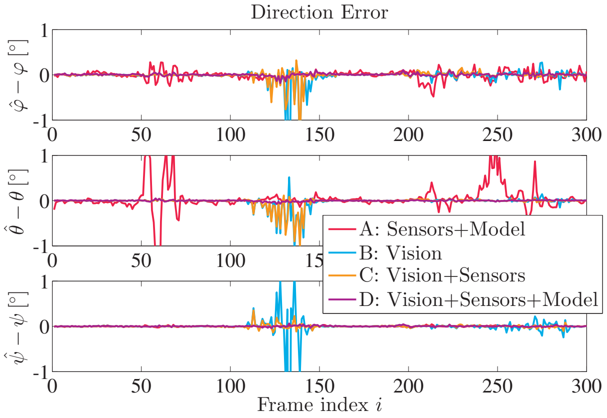

Fig. 6. Local direction estimation error between consecutive frames in the sequence. Ground truth angles are denoted φ, θ and ψ for roll, pitch and yaw respectively. Corresponding estimates are denoted $\hat{\varphi}$, $\hat{\theta}$ and $\hat{\psi}$.

adding the motion model improves all three angle estimates. The non-vision approach, i.e. algorithm A, only estimates planar motion meaning that the local errors for roll and pitch are the true roll and pitch angles with opposite sign. For instance, the large spikes in pitch error in the left part of Figure 6 show when the vehicle negotiates a speed bump.

The local errors for the position estimates are visualized in Figure 7. Worth noticing is that the algorithms utilizing the motion model, i.e. the non-vision approach A and the vision approach D, has relatively small errors compared to algorithms B and C. Algorithm D manages to combine the small errors in position and yaw angle from the non-vision algorithm A with the small errors in pitch and roll angle which are seen for most parts of the sequence for the vision algorithm B.

VII. Conclusion

We have presented an approach for 6-DoF vehicle pose estimation using a single vehicle-based camera. Visual features were complemented by standard in-vehicle sensors and a single-track vehicle model in bundle adjustment framework. The method has been tested experimentally in challenging situations from a vision perspective and in cases where the assumptions in the vehicle model are invalid, e.g. speed bumps.

The non-vision approach, meaning vehicle model and in-vehicle sensors, produces accurate pose estimates for planar vehicle motion, i.e. x, y and ψ. Vision-based approaches give more accurate direction estimates, compared to the non-vision approach. This arises because vision is the only considered input which can estimate deviations from planar motion, i.e. changes in z, φ and θ. When combining the vehicle model with vision and in-vehicle sensors, results show that the accuracy from the non-vision approach in estimating planar motion is preserved. Furthermore, the accuracy obtained from vision in direction estimation is not only maintained, but in fact further improved, primarily in sequences where the matched visual features are few.

Results indicate that using single inexpensive image sensors has the potential to produce accurate and robust vehicle pose estimates, when combined with vehicle motion models. Studying the sensitivity of the method with regards to model parameters and data sets remains for future work.

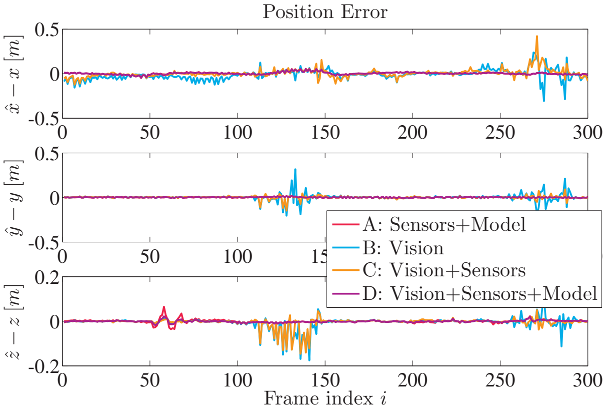

Fig. 7. Local position estimation error between consecutive frames in the sequence. Ground truth position components are denoted x, y and z while the corresponding estimates are denoted $\hat{x}$, $\hat{y}$ and $\hat{z}$.

References

[1] J. Nilsson, A. C. Ödblom, J. Fredriksson, A. Zafar, and F. Ahmed, “Performance Evaluation Method for Mobile Computer Vision Systems using Augmented Reality,” in *IEEE Virtual Reality Conference*, (Waltham, USA), pp. 19–22, IEEE, Mar. 2010.

[2] R. Azuma and Others, “A survey of augmented reality,” *Presence-Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[3] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press, 2003.

[4] C. Engels, H. Stewénius, and D. Nistér, “Bundle adjustment rules,” *Photogrammetric Computer Vision*, vol. 2, 2006.

[5] K. Konolige, M. Agrawal, and J. Sola, “Large-scale visual odometry for rough terrain,” *Robotics Research*, 2011.

[6] S. Thrun, “Robotic mapping: A survey,” in *Exploring artificial intelligence in the new millennium* (G. Lakemeyer and B. Nebel, eds.), no. February, pp. 1–35, Morgan Kaufmann, 2002.

[7] D. Nistér, O. Naroditsky, and J. Bergen, “Visual odometry for ground vehicle applications,” *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[8] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, “Real Time Localization and 3D Reconstruction,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, pp. 363–370, IEEE, 2006.

[9] J. Michot, A. Bartoli, and F. Gaspard, “Bi-objective bundle adjustment with application to multi-sensor slam,” *3DPVT’10*, 2010.

[10] M. Lhuillier, “Incremental Fusion of Structure-from-Motion and GPS using Constrained Bundle Adjustments.,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, pp. 2489–2495, July 2012.

[11] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, “Monocular visual odometry in urban environments using an omnidirectional camera,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2531–2538, IEEE, Sept. 2008.

[12] D. Scaramuzza, “Performance evaluation of 1-point-RANSAC visual odometry,” *Journal of Field Robotics*, vol. 28, no. 5, pp. 792–811, 2011.

[13] J. Kuipers, *Quaternions and rotation sequences*. Princeton, New Jersey: Princeton University Press, 1999.

[14] H. Pacejka, *Tyre and Vehicle Dynamics*. Elsevier Ltd, 2 ed., 2006.

[15] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, “Bundle Adjustment A Modern Synthesis,” *Vision algorithms: theory and practice*, vol. 34099, pp. 153–177, 2000.

[16] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, Nov. 2004.