2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

A Scalable Framework for Robust Vehicle State Estimation with a Fusion of a Low-Cost IMU, the GNSS, Radar, a Camera and Lidar

Yuran Liang[1,2], Steffen Müller[1], Daniel Schwendner[2], Daniel Rolle[2], Dieter Ganesch[2], Immanuel Schaffer[2]

Abstract— **Automated driving requires highly precise and robust vehicle state estimation for its environmental perception, motion planning and control functions. Using GPS and environmental sensors can compensate for the deficits of the estimation based on traditional vehicle dynamics sensors. However, each type of sensor has specific strengths and limitations in accuracy and robustness due to their different properties regarding the quality of detection and robustness in diverse environmental conditions. For these reasons, we present a scalable concept for vehicle state estimation using an error-state extended Kalman filter (ESEKF) to fuse classical vehicle sensors with environmental sensors. The state variables, i.e., position, velocity and orientation, are predicted by a 6-degree-of-freedom (DoF) vehicle kinematic model that uses a low-cost inertial measurement unit (IMU) on a customer vehicle. The Error of the 6-DoF rigid body motion model is estimated using observations of global position using the global navigation satellite system (GNSS) and of the environment using radar, a camera and low-cost lidar. Our concept is scalable such that it is compatible with different sensor setups on different vehicle configurations. The experimental results compare various sensor combinations with measurement data in scenarios such as dynamic driving maneuvers on a test field. The results show that our approach ensures accuracy and robustness with redundant sensor data under regular and dynamic driving conditions.**

I. INTRODUCTION

Egomotion estimation, such as longitudinal velocity and side-slip angle estimation has been an important research topic in vehicle dynamics control for decades. It is achieved by fusing vehicle dynamics sensors including wheel speed sensors, steering angle sensors and inertial sensors [1]. The requirements for the accuracy and robustness of velocity and odometry estimation has been increasing in the past ten years to enable advanced driver assistance systems (ADASs) and automated driving (AD). The traditional approaches that use vehicle dynamics sensors to approximate the vehicle dynamics [2] and/or kinematics [3] face an accuracy and integrity bottleneck. Some global navigation satellite system (GNSS)-based methods can compensate for this deficit with drift-free localization and highly precise Doppler velocity but cannot guarantee the availability and reliability in scenarios such as driving through tunnels or driving in parking garage [4]. The application of environmental sensors to AD such as visual odometry (VO), radar odometry (RO) and lidar odometry (LO) enables the redundancy of the classical state estimation methods and shows the potential to improve the accuracy and robustness of the current algorithms. However, these techniques were proposed for applications such as robots and unmanned aerial vehicles (UAVs) in simple scenarios. Integration of these methods in ADASs and AD cars is so challenging that the algorithms have to be optimized to understand large-scale traffic scenes under regular and dynamic driving maneuvers.

This paper proposes a scalable concept to fuse low-cost sensors available in ADASs and AD cars. We use the well-known dead reckoning algorithm to recursively predict the state variables: the position, velocity and quaternions of the vehicle orientation in a global coordinate system (GCS). The a-priori state is then corrected through loose coupling with the GNSS, radar, a camera and lidar. Unlike a tight coupling of all sensors, loose coupling enables the scalability of various sensor combinations, because the sensors can be configured depending on requirements for accuracy and robustness. In addition, scalability enhances the robustness in case one or more sensors fail or are not available. Furthermore, the linearity error of an ESEKF can be lower than that of a vanilla EKF since an ESEKF estimate the error state, which has a smaller degree of non-linearity and simpler dynamics, instead of the full state.

Additionally, we propose multiple sensor models for the distributed estimation. Our sensor models are egomotion estimation algorithms that use raw sensor data such as point clouds from lidar. Compared to other environmental sensors, radar has the advantage that it provides accurate relative velocity values through the Doppler effect. We propose an outlier detection algorithm with physical constraints based on the Hough transformation [5]. In the camera model, we utilize transfer learning of different deep convolutional neural network (CNN) frameworks to estimate dense optical flow for the essential matrix estimation of a mono-camera. These learning-based methods are compared with traditional dense and sparse optical flow estimations, such as the Farneback [6] and Lucas–Kanade [7] methods. For registration of lidar point clouds with iterative closet point (ICP) pipelines [8], we adopt a density-based spatial clustering of applications with noise (DBSCAN) algorithm [9] to cluster the objects in the scene. Principal component analysis (PCA) [10] is used to understand the scene and select the regions of interest (ROIs) for ICP.

In summary, we make the following contributions to vehicle state estimation:

- Enabling a generic framework to fuse sensor data for various vehicle sensor setups.
- Safety through redundancy: increasing robustness through using redundant sensor models as state obser-

[1]The authors are with Department Automotive Engineering, Faculty of Mechanical Engineering and Transport Systems, Technical University of Berlin, 10623 Berlin, Germany

[2]The authors are with BMW Group, 80788, Munich, Germany

vers.
- Integration of novel promising CNN frameworks on optical flow estimation in a traditional ESEKF.
- A method to identify and select ROIs in urban and highway environments for point cloud registration.

II. RELATED WORKS

Using environmental sensors for state estimation is a typical research topic in robotics, in which IMUs, cameras, RGB-D sensors and lidar are already widely used. In this section we focus on radar, camera and lidar sensor models and research on fusion of classical vehicle dynamics sensors with environmental sensors in ADAS and AD use cases.

A. Sensor Models for State Estimation

A popular visual odometry pipeline called ORB-SLAM2 was published by Mur-Artal et al. [11]. that provides a generic bundle adjustment-based simultaneous localization and mapping (SLAM) framework by exploiting ORB [12] features for monocular, stereo and RGB-D cameras, which allows for zero-drift localization. A large-scale direct sparse VO based on stereo [13] cameras was proposed by Wang et al. for real-time tracking and mapping with reasonable accuracy, compare to classical visual SLAM methods such as ORB-SLAM2 [11].

A notable early approach to egomotion estimation with radar was that of Kellner et al. [14]. The authors present a framework to detect stationary objects and estimate 2-D egomotion from their radial velocities and azimuth angles. Nevertheless, this algorithm has an indeterministic cycle time due to random sample consensus (RANSAC) detection. S. H. Cen et al. [15] introduce a RO approach using the power-range spectra of radar to extract landmarks and associating landmarks with unary descriptors and pairwise compatibility scores in two consecutive scans. [16] and [17] investigate self-supervised learning approaches to predict landmarks and their characteristics before associating two scans and to reduce odometry errors compared to the landmark extraction method of [15]. However, these studies do not utilize the precise relative Doppler velocity, which is the main advantage of radar and enables instantaneous estimation, in addition to lidar and a camera.

Lidar egomotion is also a popular topic, since a large number of lidar approaches are published in various online datasets. SuMa++ [18] uses a semantic map extracted by a CNN from the point cloud to filter dynamic objects. However, this algorithm is not capable for real time application. Zhang et al. [19] present a low-drift lidar SLAM pipeline in real time. The authors propose remarkable feature extraction for LO at a high frequency. This concept is, however, not suitable for LO in large scale scenes with low resolution. Because such lidar detects not enough features of Lines and edges in each layer.

B. Multimodal Sensor Data Fusion

On the fusion of multimodal sensor data, the tight coupling of sensor data has the advantage of accuracy but also entails more complexity and less flexibility. Visual-lidar odometry and mapping (VLOAM) [20] is a notable framework to fuse cameras and lidar tightly. High-frequency visual odometry is corrected and refined with low-frequency, drift-free lidar odometry. Besides, Clark et al. [21] and Wang et al. [22] investigate a tightly coupled camera+IMU odometry system using a CNN framework called Flownet [23] to extract geometric information from the camera and to deal with the sequential dynamics via long short-term memory (LSTM) cell. The main drawback of these methods is its low velocity resolution and detection failures in adverse weather. For this reason, our previous work [24] propose an end-to-end framework based on deep learning for vehicle side-slip angle estimation with a fusion of an IMU, a camera, lidar and radar. [24] is, however, computationally expensive and is proven to be over-fitting due to not enough training data.

The most similar approach to our work is [25], which presents a robust localization pipeline based on UKF. Instead of tight coupling, the authors fuse data from the IMU, GNSS, distance-measuring instruments (DMI), lidar and route network definition file (RNDF) loosely to achieve flexibility and reduce the complexity of the overall system. This concept is, however, only capable of matching lidar and RNDF point clouds when GNSS is available and stable. Combining cellular pseudoranges with lidar [4] can overcome this deficit in GNSS-challenged environments but requires a base station for sending and receiving signals. Concerning the trade-off between accuracy, robustness and scalability, we also adopt a loose coupling concept. In contrast to these studies, we use an ESEKF to estimate the error state but not the faster-changing full state. The interaction of the IMU, GNSS, radar, camera and lidar exploits the advantage of each sensor system and ensures robustness under environmental uncertainty and sensor failure.

III. TASK AND NOTATION

The task addressed in this work is vehicle state estimation with a fusion of vehicle dynamic sensors and environmental sensors to achieve signal quality with high robustness and accuracy. The problem statement is formulated as follows: Given sensors data from an IMU, the GNSS, radar, a camera and lidar, determine the vehicle positions, velocities and orientations in the global frame and velocities in the body frame.

Throughout this paper, light, bold lower-case letters and bold upper-case letters represent scalars, vectors and matrices, such as Δt, $\boldsymbol{p}$ and $\boldsymbol{P}_L$. Superscripts $-$ and $+$ denote a-priori and a-posteriori states, respectively, such as $\boldsymbol{x}^-$ and $\boldsymbol{x}^+$. The operators $\otimes$ and $[\cdot]_\times$ are the quaternion product and skew-symmetric of a vector with 3 elements, while $\boldsymbol{q}\{\cdot\}$ denotes a special orthogonal group $SO(3)$ represented as a quaternion. The identity matrix is defined as $\boldsymbol{I}$, and the subscript denotes its dimension; e.g., $\boldsymbol{I}_3$ is a 3×3 identity matrix. We use a bold zero $\mathbf{0}$ to represent the zero matrix, and its dimension depends on the dimension of the matrix that needs to be padded.

IV. FUSION FRAMEWORK BASED ON ESEKF

Fig. 1 gives a system overview of the proposed fusion framework based on an ESEKF. The state variables $\boldsymbol{x}$ are recursively predicted by the motion model. As soon as at least one observation from the observation model is available, the error state $\boldsymbol{\delta x}$ is calculated, and the predicted state $\boldsymbol{x}^-$, i.e., the a-priori state or nominal state, is updated. The corrected state $\boldsymbol{x}^+$ is also referred to a-posteriori state. In our system, we follow the ISO 8855 [26] to define the right-hand vehicle body frame; for this convention, the vehicle GCS is defined on the northwest-up (NWU) frame. The output velocities are also transformed into the body frame, since they are important physical quantities for high-level functions such as object tracking and motion control.

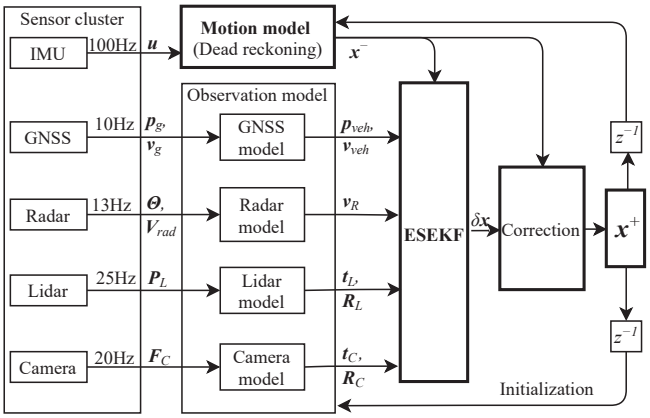

Fig. 1: System overview of the ESEKF fusion concept.

The motion model is a 6-DoF vehicle kinematic model. Its inputs $\boldsymbol{u}$ are the accelerations $\boldsymbol{u}_a$ and angular velocities $\boldsymbol{u}_\omega$ in the body frame that are obtained by an IMU, sampled at a rate of 100 Hz. The nominal state is recursively predicted if no observation is available in the estimation loop. However, due to the systematic error of the IMU, such as bias, white noise, and the linearized error of the motion model, the filter will diverge with time if the predicted state is not corrected by observation.

The observation model consists of GNSS, radar, camera, and lidar sensor models, which estimate egomotion at different rates due to the asynchronous cycle time of each sensor. In the GNSS model, we utilize positions in the geodetic coordinate system $\boldsymbol{p}_g$ and velocities in the northeast-down (NED) frame $\boldsymbol{v}_g$ from the GNSS receiver to observe odometry and velocities in the NWU frame. Radar can detect targets in the environment and estimate their positions and radial velocities relative to the sensor. As point clouds can also be gathered from lidar with higher density, in the radar model, we only focus on exploiting azimuth angles $\boldsymbol{\Theta}$ and radial relative velocities $\boldsymbol{V}_{rad}$ for estimation of longitudinal and lateral velocity $\boldsymbol{v}_R$. In addition, in the camera model, the translation and rotation of the vehicle $\boldsymbol{t}_C$ and $\boldsymbol{R}_C$ is calculated with two consecutive images from a monocular camera in a time sequence, using the estimation of the previous cycle as a scale factor. The vehicle translation $\boldsymbol{t}_L$ and rotation $\boldsymbol{R}_L$ are redundantly estimated in the lidar model with a point cloud registration technique. All the outputs of the sensor models are then treated as measurements for the ESEKF, in which the error state and its error covariance are estimated. The corrected state is then not only transferred to the next recursion but also used for the initialization of the algorithms in the observation model that stabilizes and accelerates our sensor models.

A. Motion Model

The state variables $\boldsymbol{x} = [\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}]^T$, namely position, velocity and quaternions, are estimated in the global frame. The orientation is represented in quaternions. The error state $\boldsymbol{\delta x}$ describes the error in position $\boldsymbol{\delta p}$, velocity $\boldsymbol{\delta v}$ and orientation $\boldsymbol{\delta\theta}$ in Euler angles and is propagated using the following equations [27]:

$$\begin{aligned}
\delta\boldsymbol{p} &\leftarrow \delta\boldsymbol{p} + \delta\boldsymbol{v}\Delta t - \frac{1}{2}[\boldsymbol{R}^N\boldsymbol{u}_a]_\times \Delta t^2 \delta\boldsymbol{\theta}, \\
\delta\boldsymbol{v} &\leftarrow \delta\boldsymbol{v} + (-[\boldsymbol{R}^N\boldsymbol{u}_a]_\times (\boldsymbol{I}_3 - \frac{1}{2}[\boldsymbol{u}_\omega]_\times \Delta t)\Delta t)\delta\boldsymbol{\theta} + \boldsymbol{v}_w, \\
\delta\boldsymbol{\theta} &\leftarrow \boldsymbol{R}^N\{\boldsymbol{u}_\omega \Delta t\}^T \delta\boldsymbol{\theta} + \boldsymbol{\theta}_w,
\end{aligned} \tag{1}$$

where Δt denotes the duration between two IMU measurements, $\boldsymbol{R}^N$ is the rotation matrix between body and NWU frame and $\boldsymbol{g}$ is the acceleration due to gravity. $\boldsymbol{v}_w$ and $\boldsymbol{\theta}_w$ are the process noise in velocity and orientation.

The Jacobian of the error propagation function $\boldsymbol{f} = [\delta\boldsymbol{p}, \delta\boldsymbol{v}, \delta\boldsymbol{\theta}]^T$ with respect to the error state is given by:

$$F = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\delta x}} = \begin{bmatrix} \boldsymbol{I}_3 & \Delta t & -\frac{1}{2}[\boldsymbol{R}^N\boldsymbol{u}_a]_\times \Delta t^2 \\ \boldsymbol{0} & \boldsymbol{I}_3 & -[\boldsymbol{R}^N\boldsymbol{u}_a]_\times (\boldsymbol{I}_3 - \frac{1}{2}[\boldsymbol{u}_\omega]_\times \Delta t)\Delta t \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}^N\{\boldsymbol{u}_\omega \Delta t\}^T \end{bmatrix}. \tag{2}$$

B. Measurement Update

The error state can be computed and the nominal state updated by the measurement from each sensor model. The relation between the measurement $\boldsymbol{z}$ and the state variable $\boldsymbol{x}$ is described by $\boldsymbol{z} = h(\boldsymbol{x}) + \boldsymbol{\upsilon}$, where $\boldsymbol{\upsilon}$ is the noise.

The Jacobian $\boldsymbol{H}$ of the measurement function $h(\cdot)$ with respect to the error state can be expressed using $\boldsymbol{X}_{\boldsymbol{\delta x}}$, the Jacobian of the true state $\boldsymbol{x}_t$ with respect to the error state as shown in the following equations [27]:

$$\boldsymbol{H} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\delta x}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_t} \frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{\delta x}} = \boldsymbol{H}_{\boldsymbol{x}} \boldsymbol{X}_{\boldsymbol{\delta x}}. \tag{3}$$

$\boldsymbol{X}_{\boldsymbol{\delta x}}$ is identical, while the Jacobian $\boldsymbol{H}_{\boldsymbol{x}}$ differs from sensor to sensor.

C. GNSS Model and Update

Geodetic positions $\boldsymbol{p}_g = [\lambda, \Phi, h]^T$, i.e., latitude, longitude and altitude, and NED velocities $\boldsymbol{v}_g = [v_N, v_E, v_D]^T$ are exploited in the GNSS model. The geographic positions are converted into the local tangent plane NWU. The origin of the NWU is set to the position $\boldsymbol{p}_0 = [p_{x0}, p_{y0}, p_{z0}]^T$ where the Kalman filter is initialized. The error state is then updated with the observation $\boldsymbol{z}_g = [\boldsymbol{p}_{veh}, \boldsymbol{v}_{veh}]^T$ in the NWU frame.

The measurement function of GNSS update can be described as:

$$\boldsymbol{z}_g = h_g(\boldsymbol{x}^-, \boldsymbol{v}_g) = \boldsymbol{H}_{\boldsymbol{x},g}\boldsymbol{x}^- + \boldsymbol{v}_g, \tag{4}$$

$$\boldsymbol{H}_{\boldsymbol{x},g} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_3 & \boldsymbol{0} \end{bmatrix}. \tag{5}$$

$\boldsymbol{H}_{\boldsymbol{x},g}$ is a 6×10 matrix that denotes the Jacobian of the measurement function with respect to the true state.

D. Radar Model and Update

To analyze the velocity profiles with radar, we use a physical model with azimuth angles $\boldsymbol{\Theta} = [\theta_1, ..., \theta_n]^T$ and relative radial velocities $\boldsymbol{V}_{rad}$ as inputs. As radar has a low vertical resolution and looks forward almost parallel to the ground, we do not use the elevation angles, and we reduce the dimension to a 2-D plane. This physical model describes the relationship between the sensor velocity $\boldsymbol{v}_R = [v_x^R, v_y^R]^T$ and the relative velocity of stationary targets. It is expressed by:

$$\boldsymbol{V}_{rad} = \begin{bmatrix} v_1^r \\ \vdots \\ v_n^r \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \vdots & \vdots \\ \cos(\theta_n) & \sin(\theta_n) \end{bmatrix} \begin{bmatrix} v_x^R \\ v_y^R \end{bmatrix}. \tag{6}$$

The radar velocity $\boldsymbol{v}_R$ can be solved with a least-squares solution using singular value decomposition (SVD). The physical model requires that all the reflections are stationary objects, which is uncommon in the ADAS and AD use case. Our algorithm works under the assumption that most of the reflected targets are stationary objects. Therefore, dynamic objects are treated as outliers of the measurement. We develop an outlier detection algorithm based on the Hough transformation, seeking robust detection and a deterministic cycle time. The basic idea is to convert the points from the data space into a parameter space and cast votes into the bin/container with maximum accumulation. Hough was widely used in early computer vision tasks to find lines and cycles and is also suitable for detecting targets within a certain class of shapes [5]. The curve in parameter space is nonlinear. Consequently, instead of representing it in a slope-intercept plane, we represent the curve with a normal parametrization in polar form. As shown in Fig. 2, every pair of azimuth angles and relative velocities corresponds to a point in the $\theta - v^{rad}$ data space on the left and a line in the $v_x^R - v_y^R$ parameter space on the right. Every combination of v_x^R and v_y^R is a container in the parameter space, and the bin with most accumulation has the greatest likelihood of a local maximum and is voted as an inlier while the rest are voted as outliers. As mentioned in the system overview, the previous state estimation is also used to correct and initialize the sensor model. Therefore, we utilize the velocity from the previous cycle to perform prefiltering of the outliers and initialization of the search area in the parameter space. In this way, the runtime can be reduced significantly, and the radar model is more robust in a scenario with predominantly dynamic objects.

The error state is updated with radar velocity $\boldsymbol{z}_R = [\boldsymbol{v}_x^R, \boldsymbol{v}_y^R]^T \in R^2$ in the vehicle frame. As the state variables

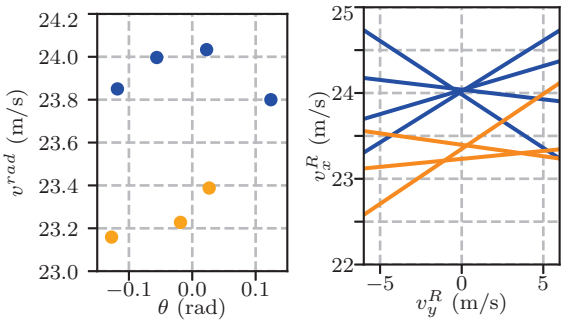

Fig. 2: Outlier detection using Hough transformation with a normal parametrization in a polar form.

are defined in the NWU frame, the measurement model with radar in the ESEKF is then derived by:

$$\boldsymbol{z}_R = h_R\left(\boldsymbol{x}^-, \boldsymbol{v}_R\right) = \boldsymbol{R}_{2\times3}^{N\,-}\boldsymbol{v}^- + \boldsymbol{v}_R. \tag{7}$$

$\boldsymbol{R}_{2\times3}^{N\,-}$ is a 2×3 matrix that is obtained by the first two rows of the inverse of the predicted rotation matrix $\boldsymbol{R}^{N-}$. The part of the Jacobian with respect to the true state $\boldsymbol{H}_{\boldsymbol{x},R}$ is a 2×10 matrix and is represented as follows:

$$\boldsymbol{H}_{\boldsymbol{x},R} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{R}_{2\times3}^{N\,-} & \boldsymbol{H}_Q \end{bmatrix}, \tag{8}$$

where $\boldsymbol{H}_Q$ is a 2×4 matrix and denotes the Jacobian of $\boldsymbol{R}_{2\times3}^{N\,-}\boldsymbol{v}^-$ with respect to the quaternions.

E. Camera Model and Update

Our camera egomotion pipeline, as shown in Fig. 3, exploits direct visual odometry (DVO) using a PWC Net [28] to extract the dense optical flows of two consecutive images F_1 and F_2 from a monocular camera. The essential matrix E of the camera is then computed with the estimated flow, given the intrinsic parameter. Since the algorithm works under the assumption that most of the pixels represent stationary objects, outlier detection is also carried out in the essential matrix estimation. The uncertainty of the scale factor can be estimated by various external sources, such as the GNSS, the radar and lidar model, the wheel speed sensors and the previous estimation of the ESEKF. The rotation matrix R and translation t are calculated through SVD. The translation is then reconstructed with the external scale factor, since the mono-camera cannot provide deep information according to its pinhole model.

Optical flow estimation is a key topic in VO, since the accuracy and robustness of VO depends on the end-point error (EPE) of optical flow. The principle of optical flow obeys the assumption that the intensity of targets is assumed to be constant and only the positions of pixels change when the targets move. In this work, we carry out transfer learning with a pretrained CNN called PWC Net [28] and adapt it to match our camera data. We try different state-of-the-art optical flow methods and compare them in terms of accuracy and runtime of the camera model. PWC Net is proven to be the most suitable framework with a reasonable computational requirement in our approach.

Fig. 4 shows a boxplot to compare the absolute position error (APE) of the camera model based on a sparse

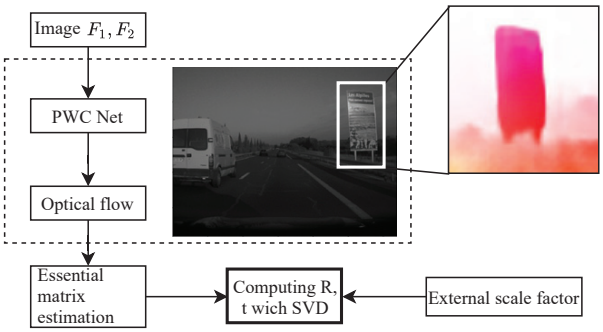

Fig. 3: Estimation of egomotion through optical flow with two consecutive images and an external scale factor.

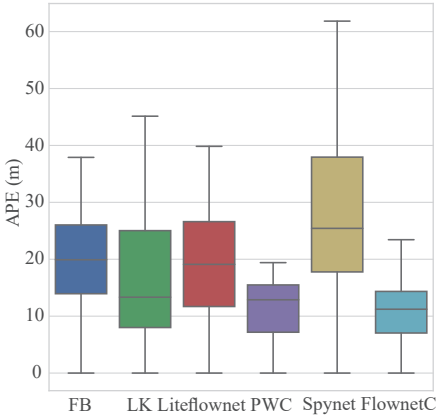

Fig. 4: Comparing the absolute position error (APE) of various optical flow approaches—Farneback (FB), Lucas-Kanade (LK), Liteflownet, PWC Net, Spynet and FlownetC—in a boxplot.

optical flow approach, the Lukas-Kanade method; a dense algorithm, Farneback; and various deep learning approaches, e.g., Liteflownet [29], PWC Net [28], Spynet [30] and FlownetC [23]. The dense optical flow methods except Spynet have the advantage in the standard deviation over the sparse concept, because the sparse concept is not as robust against outliers in the flow as the dense optical flow, which considers much more semantic information in the scene. The learning-based models except Spynet have not only the benefit of robustness but also accuracy with a lower median and a smaller variance. PWC Net and FlownetC have comparable performance; however, FlownetC requires much more memories for deployment. Therefore, we chose PWC Net as our flow estimator in the camera model. The output 3×3 rotation matrix $\boldsymbol{R}_C$ is converted to quaternions for a convenient derivation of the measurement function. Regarding the robustness of the camera model, $\boldsymbol{t}_C$ is reduced to 2-D and converted to velocities using the time difference, i.e., $\boldsymbol{v}_C = [v_x^C, v_y^C]^T$, in the body frame. The observation $\boldsymbol{z}_C = [\boldsymbol{v}_C, \boldsymbol{q}_C]^T$ is then utilized to update the error state expressed by:

$$\begin{aligned} \boldsymbol{z}_C &= h_C\left(\boldsymbol{x}^-, \boldsymbol{v}_C\right), \\ \boldsymbol{v}_C &= \boldsymbol{R}_{2\times 3}^{N\ -} \boldsymbol{v}^- + \boldsymbol{v}_{C,v}, \\ \boldsymbol{q}_C &= \boldsymbol{q}^- \otimes \boldsymbol{v}_{C,q}. \end{aligned} \tag{9}$$

The 6×10 Jacobian with respect to the true state $\boldsymbol{H}_{\boldsymbol{x},C}$ is given by:

$$\boldsymbol{H}_{\boldsymbol{x},C} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{R}_{2\times 3}^{N\ -} & \boldsymbol{H}_Q \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_4 \end{bmatrix}. \tag{10}$$

F. Lidar Model and Update

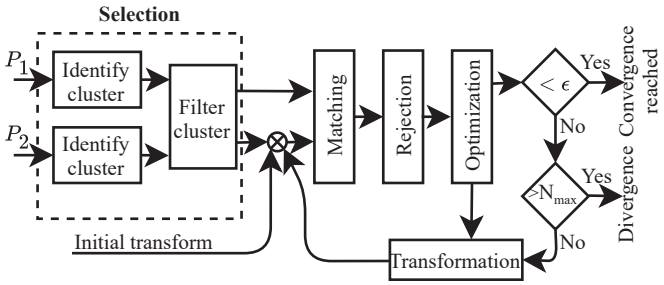

Fig. 5: Overview of the lidar egomotion model.

The lidar egomotion framework is shown in Fig. 5, and it estimates the optimal transformation between two consecutive point clouds P_1 and P_2. The first step is to filter adverse objects, namely dynamic objects such as other vehicles and unwanted static objects such as road barriers, which are treated as perturbations in the point cloud registration. Noisy objects that change in size or shape between two scans are also deselected. This is achieved by using DBSCAN for clustering the point cloud and PCA for classifying objects. Corespondent search is performed using the Euclidean distance metrics and a kd-tree structure for the nearest-neighbor search [8]. To improve computational efficiency and accuracy, an initial transform is applied using the previous ESEKF estimation. If the distance between two matched points exceeds a threshold, the pair is rejected. Finally, the optimal transformation is calculated by minimizing the error between the two point clouds using SVD. The matching, rejection and optimization steps are repeated until the error is below a certain threshold ε or a maximum number of iterations N_{max} is exceeded.

The ICP algorithm provides a precise estimation only if the environment is static and contains only rigid and stationary objects. Particularly for low-resolution lidar in automotive applications, this assumption cannot be met. An example is therefore shown in Fig. 6, which displays two consecutive point clouds during a highway driving scenario. The vehicle in front is detected by the lidar in the form of a horizontal line of points referring to cluster 4. This dynamic object is an outlier and it is thus filtered according to the singular value from the PCA. A second type of region to be filtered contains objects represented by long longitudinal lines of points referring to cluster 10, as localization in the x-direction is challenged. Since they provide the similar appearance over time. These can include road barriers and tunnel walls. In noisy environments, the shape and number of points belonging to the same object can vary between two scans referring to cluster 8 and 9, resulting in faulty correspondence search. Thus, they are treated as outliers. These outliers are identified by means of analyzing their singular values.

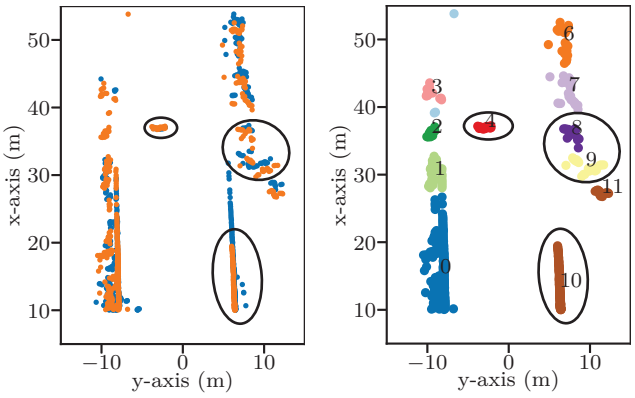

Fig. 6: Clustering of point clouds with DBSCAN and filtering with PCA. The current (orange) and previous (blue) point cloud contain various unwanted targets marked with ellipses.

Similar to the camera model, the transformation is observed by the lidar model. The measurements $\boldsymbol{t}_L$ and $\boldsymbol{R}_L$ from the lidar model are transformed into velocities $\boldsymbol{v}_L$ in the body frame and quaternions $\boldsymbol{q}_L$. Consequently, the lidar update is computed as follows:

$$\begin{aligned} \boldsymbol{z}_L &= h_L\left(\boldsymbol{x}^-, \boldsymbol{v}_L\right), \\ \boldsymbol{v}_L &= \boldsymbol{R}^{N-}\boldsymbol{v}^- + \boldsymbol{v}_{L,v}, \\ \boldsymbol{q}_L &= \boldsymbol{q}^- \otimes \boldsymbol{v}_{L,q}. \end{aligned} \tag{11}$$

The Jacobian with respect to the true state $\boldsymbol{H}_{\boldsymbol{x},L}$ is a 7×10 matrix and is given by:

$$\boldsymbol{H}_{\boldsymbol{x},L} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{R}^{N-} & \boldsymbol{H}_Q \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_4 \end{bmatrix}. \tag{12}$$

The noise variance of rotation is set to a large value, since our lidar model cannot estimate the rotational motion with sufficient quality. Due to the low vertical resolution of the lidar sensor, the lidar model is not suitable for motion estimation in the z direction. We see further potential in the improvement of lidar egomotion and are addressing this in future work.

V. EXPERIMENTAL RESULTS

Our proposed ESEKF fusion concept has been evaluated in different scenarios. In our test, we carry out diverse open-loop and closed-loop handling maneuvers, which are typical for the testing of chassis control systems. The algorithm has been evaluated using absolute error analysis with various metrics.

A. Sensor Setup for the Experiment

In the test vehicle, we use the following sensors for the fusion method:

- a low-cost IMU with a 100 Hz sampling rate from a BMW vehicle Flexray-Bus,
- a GNSS receiver with a 10 Hz measurement rate with a ublox evaluation kit,
- a grayscale camera with a 20 Hz sampling rate in a BMW vehicle setup (front-middle),
- a long-range radar with a 13 Hz sampling rate in a BMW vehicle setup (front-middle),
- a 4-layer lidar with a 25 Hz sampling rate (front-middle).

In addition, a highly precise automotive dynamic motion analyzer (ADMA) with double antenna based on an IMU and Differential Global Positioning System (DGPS) has been implemented in the test vehicle to generate the ground truth of egomotion. The ADMA offers highly accurate vehicle orientation, velocity and position data (2 cm).

The extrinsic parameters between the sensors and vehicle rear axis have already been calibrated. The intrinsic parameter as well as the distortion model and its parameters for camera are also calibrated.

B. Quantitative Evaluation and Discussion

We evaluate the absolute position error (APE) and absolute vehicle body velocity error (AVE) of our ESEKF compared to the ground truth. Firstly, our approach is evaluated with several typical maneuvers for testing of vehicle dynamics. Table I shows the APE and AVE in different scenarios, namely, multi-circular driving on a low-friction road (scenario 1), drift on a low-friction road (scenario 2), dynamic driving on a handling course with disturbed GNSS signals (scenario 3), circular driving on a road with a large slope (scenario 4) and sinusoidal steering with different velocities on a test field (scenario 5). We use the following evaluation metrics: root-mean-square error ε_{RMSE}, mean error μ, median error ε_{Q50}, standard deviation σ and maximum error ε_{MAX}. The travelled distance d of each maneuver is also given. The ESEKF performs best on low dynamic driving (scenario 1), with the lowest APE and AVE with respect to all metrics. Even with highly dynamic driving (scenario 2), the ESEKF still has promising performance in velocity estimation. The accuracy is reduced if the effects of external disturbance or model imperfection are no longer negligible, such as disturbed GNSS signals, large excitation on the road and fast-changing system dynamics, due to the linearized error of the Kalman filter. The estimation of velocity shows more robustness compared to odometry, since we do not observe the vehicle positions with environmental sensors directly, but only the velocities. Since the noise variance of the roll and pitch angle in the measurement models are not yet optimized, the rotations are not properly estimated, which leads to a reduced signal quality on the large road slope.

The fine-tuning of all parameters in the Kalman filter will be carried out in our future work.

The performance of the observations with an individual sensor is summarized in Fig. 8 with boxplots of the APE and AVE. The ESEKF is evaluated in the multi-circular driving with stable GNSS signals. The GNSS has a significant advantage in low-drift odometry estimation in comparison to environmental sensors due to its precise satellite-based localization, followed by radar, camera and lidar. For the velocity estimation in the body frame, all the sensors reach a median absolute error under $2m/s$. GNSS and radar models are capable of estimating the instantaneous velocity with a Doppler effect and are consequently advantageous for velocity estimation, while camera and lidar models compute

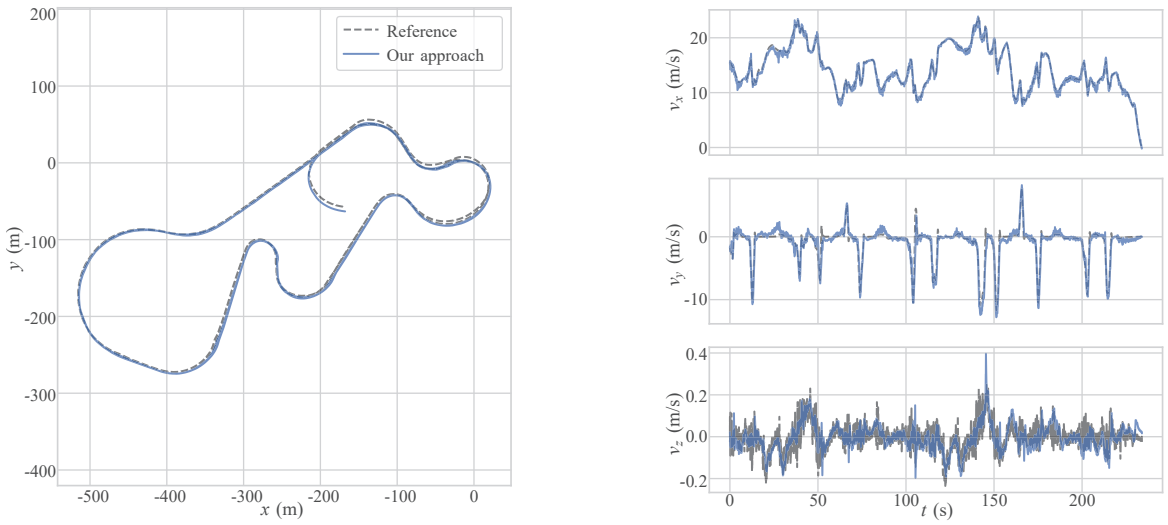

Fig. 7: 2D trajectory and 3D velocities of our approach compared with a ground truth reference in highly dynamic driving condition on low-friction roads.

TABLE I: Evaluation of APE and AVE for various driving maneuvers.

	d(m)		ε_{RMSE}	μ	ε_{Q50}	σ	ε_{MAX}
Szenairo 1	2247	APE (m)	0.91	0.83	0.89	0.36	1.72
		AVE (m/s)	0.71	0.59	0.50	0.39	2.75
Szenairo 2	4021	APE (m)	2.81	2.25	1.61	1.68	5.26
		AVE (m/s)	0.71	0.60	0.56	0.39	2.46
Szenairo 3	4574	APE (m)	3.38	2.98	3.45	1.59	5.36
		AVE (m/s)	0.67	0.51	0.37	0.43	2.79
Szenairo 4	1532	APE (m)	4.31	3.93	4.87	1.78	6.06
		AVE (m/s)	0.84	0.71	0.66	0.46	3.49
Szenairo 5	2186	APE (m)	5.42	5.13	5.74	1.76	7.72
		AVE (m/s)	1.30	0.96	0.65	0.88	4.45

the velocity using two consecutive frames within a time duration. The radar ESEKF demonstrates the best accuracy on velocity with the lowest median and box width, since the radar model calculates the local sensor velocity and the GNSS model has to convert its Doppler velocity from the global frame into the local frame, which needs to determine the heading angle.

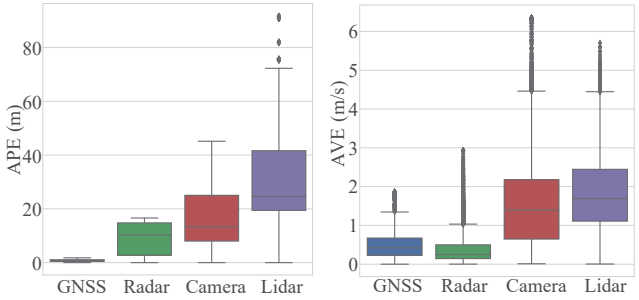

Fig. 8: Performance analyze of an observer with an individual sensor in terms of APE and AVE.

In order to evaluate the scalability of our algorithm, we compare the performance of the different sensor setups in Fig. 9. The absolute position error of the basic setup with the GNSS oscillates in a wider range than the fusion with environmental sensor data. Zooming in on Fig. 9 (a) on the right, Fig. 9 (b), shows a situation where the error of the GNSS fusion is increasing, which means that the error state is not properly estimated by the GNSS due to its localization error. The error caused by GNSS observation can be incrementally compensated for with radar, camera and lidar. Therefore, the robustness of the system is improved with redundant environmental sensor data. Furthermore, the sensor hardware and software can then be deployed in a vehicle, depending on the requirements for accuracy and robustness, which means that our approach is scalable for different sensor setups.

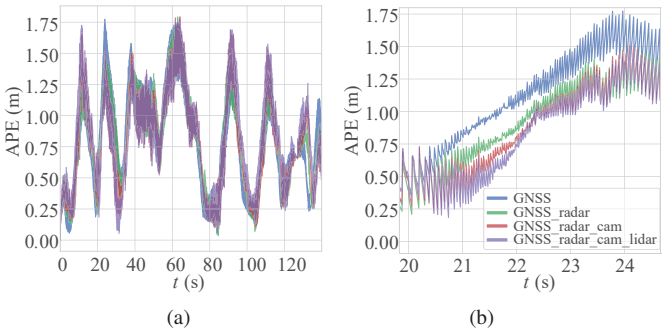

Fig. 9: Incremental sensor setups for the sensor data fusion.

At the end of the evaluation, Fig. 7 shows a comparison of our approach to the ground truth in a 2D trajectory and 3D velocities in the body frame. The experiment was carried out on a test track under a drift maneuver with 4021m traveled distance and reached a maximum absolute lateral velocity about 11.75m/s. The estimated trajectory follows the ground truth well, although the odometry deviates slightly from the ground truth when drifting due to the slip. Velocities are well estimated due to the redundancy of sensor models. However, the outputs from the ESEKF are still slightly noisy. The lateral velocity is overestimated if these values are close to zero, i.e., if the vehicle is driven straight. This is mainly due to the heading offset of the GNSS, which we have not yet considered in the model.

In summary, with all the experimental results, the GNSS and radar have more potential to increase the accuracy of velocity estimation, while the GNSS has a particular advantage for odometry observation. Moreover, using environmental sensors such as radar, cameras and lidar can improve odometry estimation, compensating for the unstable GNSS performance.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we showcase an ESEKF-based fusion concept to research accurate and robust state estimation in automotive applications. The sensor models are computed with a distributed solution using loose coupling, which enables the scalable and independent development of each sensor model. The method of using PWC Net-driven camera egomotion and the outlier detection of optical flows shows a remarkable improvement compared with other state-of-the-art optical flow methods. Furthermore, the Hough transformation-based radar model enables a precise instantaneous velocity estimation that provides redundancy for the GNSS velocity. Moreover, the DBSCAN- and PCA-driven ICP in the lidar model enables a robust lidar egomotion in a complex scene. With the versatile evaluation of our system, our approach is shown to achieve not only a precise but also a robust vehicle state estimation for ADASs and AD use cases.

In future work, we intend to overcome the deficits of our approach mentioned in the last section. In particular, the optimization of the noise variants considering the conflicting objectives of different sensor models will be addressed in future work. In addition, we propose to optimize the lidar model for a better estimation of the lateral and rotational motion. A benchmark with a high-resolution lidar should be carried out to analyze the potential of our low-resolution lidar egomotion. Last, the analysis and decoupling of the sensor error and sensor model error using simulation data, as well as the prototype verification of the fusion, is necessary and will be addressed in our next work.

REFERENCES

[1] A. Brunker, T. Wohlgemuth, M. Frey, and F. Gauterin, "Odometry 2.0: A slip-adaptive eif-based four-wheel-odometry model for parking," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 114–126, 2018.

[2] D. Selmanaj, M. Corno, G. Panzani, and S. M. Savaresi, "Robust vehicle sideslip estimation based on kinematic considerations," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 855–14 860, 2017.

[3] J. Bechtoff and R. Isermann, "Cornering stiffness and sideslip angle estimation for integrated vehicle dynamics control," *IFAC-PapersOnLine*, vol. 49, no. 11, pp. 297–304, 2016.

[4] M. Maaref, J. Khalife, and Z. M. Kassas, "Lane-level localization and mapping in gnss-challenged environments by fusing lidar data and cellular pseudoranges," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 73–89, 2018.

[5] N. Kiryati, Y. Eldar, and A. M. Bruckstein, "A probabilistic hough transform," *Pattern recognition*, vol. 24, no. 4, pp. 303–316, 1991.

[6] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian conference on Image analysis*. Springer, 2003, pp. 363–370.

[7] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[8] W.-S. Choi, Y.-S. Kim, S.-Y. Oh, and J. Lee, "Fast iterative closest point framework for 3d lidar data in intelligent vehicle," in *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 2012, pp. 1029–1034.

[9] L. Meng'Ao, M. Dongxue, G. Songyuan, and L. Shufen, "Research and improvement of dbscan cluster algorithm," in *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*. IEEE, 2015, pp. 537–540.

[10] I. T. Jolliffe, "Principal components in regression analysis," in *Principal component analysis*. Springer, 1986, pp. 129–155.

[11] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[13] R. Wang, M. Schworer, and D. Cremers, "Stereo dso: Large-scale direct sparse visual odometry with stereo cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3903–3911.

[14] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using doppler radar," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 869–874.

[15] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[16] R. Aldera, D. De Martini, M. Gadd, and P. Newman, "Fast radar motion estimation with a learnt focus of attention using weak supervision," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1190–1196.

[17] D. Barnes and I. Posner, "Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar," *arXiv preprint arXiv:2001.10789*, 2020.

[18] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4530–4537.

[19] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.

[20] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: low-drift, robust, and fast," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2174–2181.

[21] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[22] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018.

[23] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[24] Y. Liang, S. Müller, D. Rolle, D. Ganesch, and I. Schaffer, "Vehicle side-slip angle estimation with deep neural network and sensor data fusion," in *10th International Munich Chassis Symposium 2019*. Springer, 2020, pp. 159–178.

[25] X. Meng, H. Wang, and B. Liu, "A robust vehicle localization approach based on gnss/imu/dmi/lidar sensor fusion for autonomous vehicles," *Sensors*, vol. 17, no. 9, p. 2140, 2017.

[26] "Road vehicles–vehicle dynamics and road-holding ability–vocabulary," International Organization for Standardization, Geneva, CH, Standard, 2011.

[27] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[28] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[29] T.-W. Hui, X. Tang, and C. Change Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8981–8989.

[30] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4161–4170.