Estimating the Driving State of Oncoming Vehicles From a Moving Platform Using Stereo Vision

Alexander Barth and Uwe Franke

***Abstract*—A new image-based approach for fast and robust vehicle tracking from a moving platform is presented. Position, orientation, and full motion state, including velocity, acceleration, and yaw rate of a detected vehicle, are estimated from a tracked rigid 3-D point cloud. This point cloud represents a 3-D object model and is computed by analyzing image sequences in both space and time, i.e., by fusion of stereo vision and tracked image features. Starting from an automated initial vehicle hypothesis, tracking is performed by means of an extended Kalman filter. The filter combines the knowledge about the movement of the rigid point cloud's points in the world with the dynamic model of a vehicle. Radar information is used to improve the image-based object detection at far distances. The proposed system is applied to predict the driving path of other traffic participants and currently runs at 25 Hz (640 × 480 images) on our demonstrator vehicle.**

***Index Terms*—Image sequence analysis, Kalman filtering, object detection, sensor data fusion, stereo vision.**

I. Introduction

FUTURE driver-assistance and safety systems aim to support the driver in increasingly more complex driving situations, allowing for safe and stress-free driving. Fast and reliable knowledge of the moving objects' location, as well as their movement patterns relative to the ego-vehicle, will be an essential basis for such systems.

Next-generation collision-avoidance systems have to predict where other traffic participants will be at a certain point in time to warn the driver in advance if his current driving maneuver may conflict with the driving path of others. Demands for accuracy and robustness are increasing as autonomous or partly autonomous intervention actions are involved, in particular, automated emergency braking. However, even warning systems must not produce too many false alarms.

In recent years, stereo vision systems have found their way into cars for various applications, such as lane recognition, obstacle detection, and traffic sign recognition. Throughout this paper, we will present a new approach for estimating the pose and the full motion state of detected vehicles, including velocity, acceleration, and yaw rate from a moving platform using stereo vision and optical flow. Based on the estimated motion state, it is possible to accurately predict the driving path of other traffic participants. Although this approach is generic, with the ability to track vehicles in any arbitrary direction, we will focus on oncoming traffic in this paper, whereas most existing approaches only consider vehicles driving in the same direction as the ego-vehicle.

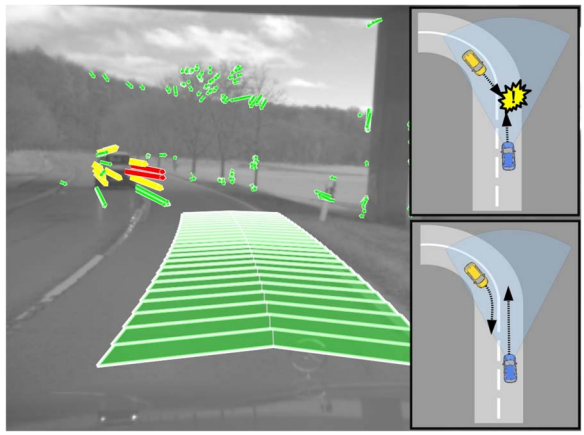


Fig. 1. Linear prediction of the driving path of an oncoming vehicle leads to false alarms in curves or turning maneuvers. Using an estimate of the other vehicle's yaw rate, it is possible to predict its real driving path accurately and to prevent such false alarms.

Estimating the driving path of oncoming vehicles on country roads is extremely challenging due to relative velocities of 100–240 km/h, narrow curved roads, and limited range of sight. In everyday situations, approaching vehicles pass very closely, making it difficult to identify real critical situations. In curves or turning maneuvers, an estimate of the yaw rate of other traffic participants is essential, as shown in Fig. 1. Using a common linear motion model, a collision between an oncoming vehicle and the ego-vehicle would be predicted. Only when the yaw rate of the oncoming vehicle is estimated, in fact, can the situation be correctly interpreted as uncritical.

Another challenging area in this field of research is environment perception at intersections from a moving platform due to the complexity of traffic. Intersections are often accident hotspots. Predicting the position of other traffic participants based on their motion pattern is an important milestone for an *intelligent* vehicle in *understanding* the complex interactions in everyday traffic situations.

This paper extends the work presented in [1]. In the proposed method, the objects are modeled by a rigid 3-D point cloud. This point cloud is tracked in an image sequence in terms of feature displacements and depth measurements from the stereo system. An extended Kalman filter is then used to relate the observations in the image to a 3-D object movement. Since the vehicle movements are physically constrained, it is possible to restrict the tracked motion to a circular path model based on

Manuscript received August 29, 2008; revised January 28, 2009. First published September 1, 2009; current version published December 3, 2009. The Associate Editors for this paper were B. DeSchutter and S. Schladover.

The authors are with the Group Research and Advanced Engineering, Daimler AG, 71059 Sindelfingen, Germany (e-mail: alexander.barth@daimler.com; uwe.franke@daimler.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2009.2029643

velocity and yaw rate. An estimate of the yaw rate of oncoming vehicles is the main new aspect of our contribution.

Three-dimensional point tracks are used to generate initial object hypotheses. The underlying idea is that a group of points within a local neighborhood moving with equal velocity in the same direction is assumed to belong to the same object. Segmenting objects in dense traffic scenes, where this condition is not satisfied, is beyond the scope of this paper. Partial occlusions, however, can occur and are discussed in Section IV-B. In addition, we will show how fusion with a radar sensor improves the vision-based object tracking, particularly on country roads. Many existing fusion approaches use vision only for the verification of radar or lidar object tracks. In our approach, we combine the advantages of the novel image-based motion estimation approach with the larger distance range of sight and accurate relative velocity measurements of the radar.

The remainder of this paper is organized as follows: Section II gives an overview on the literature related to our work. In Section III, the detection and tracking approach is introduced, which includes object, system, and measurement models, as well as the automated object hypothesis generation used for initialization. It is assumed that the reader is familiar with the basic concepts of the extended Kalman filter. For more details, please see, e.g., Bar-Shalom [2] and Simon [3]. Both simulated and real-world experiments are presented and discussed in Section IV. Section V summarizes the main results and gives an outlook on future work.

II. Related Work

The detection and tracking of vehicles has been explored by many researchers in the computer-vision community over the past two decades. Many approaches focus on traffic surveillance on highways or at intersections using static cameras placed at elevated positions [4]–[10]. These approaches first usually segment moving objects from a static background by some sort of background subtraction. Then, the detected foreground regions are analyzed and tracked. In [4], Koller *et al.* discriminate moving objects from the background on the basis of image flow and combine a 3-D vehicle motion model with a contour-based fitting of a 3-D shape model to track road traffic.

Beymer *et al.* [6] introduced a model-free object representation based on groups of corner features to yield more stable tracking results in dense traffic situations. Using this approach, single features on the object's surface are tracked instead of tracking the object as a whole, thus leading to more accurate tracking results at partial occlusions that can occur in real traffic situations, although the camera is elevated.

If the camera is mounted in a car, then many *a priori* constraints introduced in terms of static camera setups do not hold as the ego-vehicle is driving through an unknown area. Depending on the ground level and the driving maneuver, camera movements can have up to six degrees of freedom. Thus, image-based methods for compensating the motion of the ego-vehicle have been proposed [11], [12] to distinguish static from independently moving points in the scene.

From the early publications to the present, most of the research on vision-based vehicle detection and tracking from a moving platform addresses tracking leading vehicles on highways [13]–[22]. These approaches consider, for example, image statistics (edges, grey-level or color histograms, symmetry, optical flow, contours, etc.), template matching, classification, and combinations of the former to detect and track vehicles in an image. All these methods can be realized with a single camera. In this case, the distance to an object is often estimated based on the object's base point on the ground plane assumed to be known from camera calibration.

Other methods fuse monocular vision with active sensors like radar or lidar to extract more reliable information on depth, for example, [21], [23], and [24]. In such approaches, the image information is usually used for the verification or stabilization of active sensor targets.

Alternatively, stereo vision is used for depth estimation. Van der Mark and Gavrila [25] provide a good overview on stereo vision in the intelligent vehicle domain, including an extensive evaluation of different real-time stereo implementations. In many stereo-vision-based object detection approaches, the objects are modeled as upright planes on a ground plane. Such planes can be identified, for example, based on the accumulation of equal distances (or disparities) within an image region, as in [26]. The ground plane (road) does not necessarily have to be flat. A solution for dealing with nonflat roads using the so called *v-disparity* images has been proposed in [27]. Toulminet *et al.* [17] present a method that combines stereoscopic and monocular cues for vehicle detection. Stereo-only methods cannot distinguish between static obstacles and moving objects without further tracking. Furthermore, near objects are likely to be merged. Thus, methods fusing the depth information from stereo with motion have been proposed, for example, in [28] and [29].

In many approaches, *tracking* is related to rediscovering an image region labeled as *vehicle* in the next frame, such as in [30]. However, if one wants to be able to predict the other traffic participant's trajectories, one has to estimate the motion state of an observed object based on a 3-D motion model. Dellaert and Thorpe [15] use such a 3-D motion model to predict the position and shape of a bounding box model in the image with an extended Kalman filter. The yaw rate of tracked vehicles is assumed to be approximately zero. An approach for estimating the pose and linear motion state of a cuboid attached to a cluster of 3-D points is presented in [31]. Dang *et al.* [29] apply an extended Kalman filter to track a rigid point cloud from a moving platform and extract the 3-D rotation and translation parameters of this point cloud. In addition to the rigidity of the point cloud, no other constraints regarding shape or motion are defined in this approach. In practice, the vehicle movements are physically constrained. Cars cannot drive sideways. One common motion model approximating the driving path of cars, motorbikes, and vans is the well-known *bicycle model* [32]. This model assumes that the vehicles are driving on circular paths, where the radius is related to their vehicle velocity and yaw rate.

Recently, the so-called *track-before-detect* methods have attracted attention. In [33], Franke *et al.* present a system called *6D-Vision*, which fuses stereo and motion by means of Kalman filters for tracking 3-D points from a moving platform. Here, 3-D points are tracked before being grouped to semantic

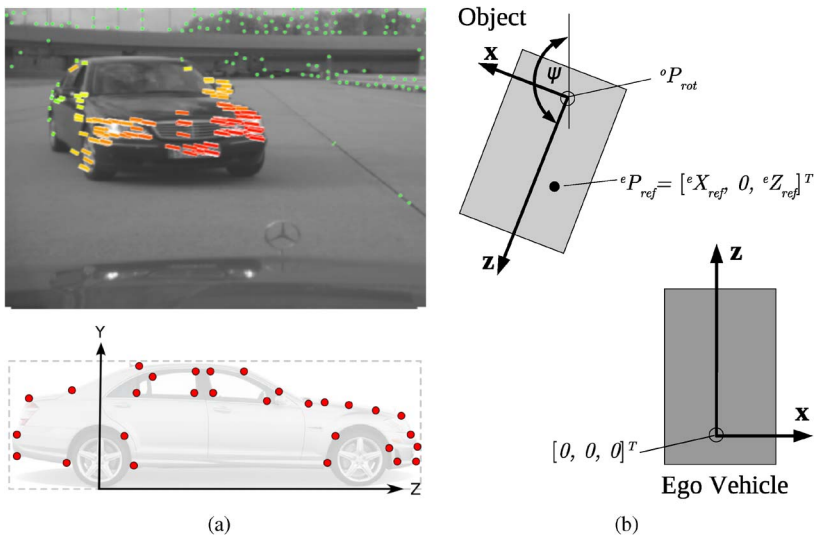


Fig. 2. (a) (Top) Points on the front surface of the vehicle show larger lateral displacements compared with points on the rear surface as the oncoming vehicle turns. (Bottom) An object's shape is modeled by a rigid point cloud. Each point has a fixed position with respect to a local object coordinate system. (b) Object and ego coordinate systems from bird's-eye view are shown. The object origin is ideally defined on the center rear axis.

objects. The filtered point tracks provide strong information for object detection, which is significantly beyond the capabilities of stereo-only or purely flow-based object detection methods.

To the authors' knowledge, there is no literature explicitly addressing the motion state estimation of oncoming traffic, including the yaw rate, from a moving platform with a stereo vision sensor. However, with respect to driver-assistance and safety systems, e.g., [34] and [35], or, in particular, on country roads and at intersections, the knowledge of where an oncoming vehicle will be the next second is highly advantageous.

III. Approach

Vehicles can be approximated by a set of 3-D points, with each point having a fixed position on the vehicle's surface. Points at different locations on the vehicle show different displacements as the vehicle moves. At the same time, the relative distance between these points stays constant (rigid body assumption). For example, for an oncoming vehicle, points at the front surface have larger lateral displacements compared with points located at the rear surface, as the vehicle is driving on a circular path [see the top part of Fig. 2(a)]. These displacements in the 3-D world can be observed in terms of image displacements and depth changes in a sequence of stereo images. Now, the task is to solve for the inverse problem, i.e., given a number of image displacements and distance measurements, find the corresponding object movement in 3-D that best matches these observations.

A. Object Model

In our approach, a vehicle is modeled by three properties:

1) pose (relative orientation and translation to ego-vehicle);
2) motion state (velocity, acceleration, and yaw rate);
3) shape (rigid 3-D point cloud).

For each vehicle, a local *object coordinate system* is defined with the origin at the center rear axis on the ground plane (street). This origin plays an important role in the system dynamics for rotational movements. x, y, and z represent the lateral axis, the height axis, and the longitudinal axis, respectively (see Fig. 2). The main moving direction is along the longitudinal axis. We will denote the coordinate system of the observing vehicle as the *ego* system. It should also be noted that all coordinate systems are left handed.

The transformation between the ego and the object's coordinate system indicates the object's pose. Motion parameters include velocity and acceleration in the longitudinal direction of the vehicle and the yaw rate. A set of M 3-D points in object coordinates ideally represents the object's shape and dimension [see the bottom part of Fig. 2(a)]. The advantage of this model is that it can be used for a wide range of variously shaped vehicles with no special adaptation.

If the point cloud model is complete and accurate, then the object's dimension can directly be derived from its bounding box. In practice, we have to deal with incomplete object models and outliers, as will be discussed in more detail in Sections III-D and E. For now, we assume the correct 3-D shape model to be known *a priori*.

B. Object Tracking

An extended Kalman filter is used for object tracking and estimation of the pose and motion parameters.

1) State Vector: The state vector for an object instance O is defined as

$$\boldsymbol{x} = \left[{}^eX_{\mathrm{ref}}, {}^eZ_{\mathrm{ref}}, {}^oX_{\mathrm{rot}}, {}^oZ_{\mathrm{rot}}, \psi, v, \dot{\psi}, a\right]^{\mathsf{T}} \tag{1}$$

where ${}^eP_{\mathrm{ref}} = [{}^eX_{\mathrm{ref}}, 0, {}^eZ_{\mathrm{ref}}]^{\mathsf{T}}$ represents the *reference point* in ego coordinates on the ground plane. The reference point does not necessarily have to correspond to the center rear axis, which is usually not known at initialization. The position of the *rotation point* at the center rear axis is given by ${}^oP_{\mathrm{rot}} = [{}^oX_{\mathrm{rot}}, 0, {}^oZ_{\mathrm{rot}}]^{\mathsf{T}}$ in object coordinates. This allows for defining the object coordinate system at an arbitrary point, e.g., the centroid of the initial point cloud [see Fig. 2(b)].

The object orientation is described by the rotation angle ψ (in radians) around the height axis between ego and object coordinate system. The object's yaw rate is given by $\dot{\psi}$ (in radians per second), and the absolute object velocity v (in meters per second) is defined along the object's longitudinal axis with acceleration a (in meters per square second). An estimate of the unknown real object state $\boldsymbol{x}$ will be denoted as $\widehat{\boldsymbol{x}}$ in the following.

2) Dynamic Model: The dynamics of a vehicle can be approximated by a bicycle motion model [32]. Lateral movements are restricted to circular path motion. Here, a simplified version, as proposed in [36], is used. Both approximately constant acceleration and yaw rates are assumed. Using the dynamic model, it is possible to predict the image position and stereo disparity for each object point ${}^{o}P_m$ for a given time step t.

Movements can be thought of as first moving the object with respect to its object coordinate system and then transforming the new position back into the ego coordinate system. Since the ego-vehicle can also move, one has to compensate for the ego-motion. One can use the same motion model for the ego-vehicle, given the ego-motion obtained from inertial sensors (velocity and yaw rate) or from highly accurate image-based methods, as proposed in [11], [12], and [37]. For simplicity, we will assume a static observer in the following, which is equivalent to an ego-motion-compensated moving platform. In practice, the ego-motion estimate is not error free; thus, the uncertainties of the ego-motion must be incorporated into the Kalman filter system noise matrix.

In general, the transformation of any object point ${}^{o}P$ is given by a rotation around ${}^{o}P_{\mathrm{rot}}$ plus the translation ${}^{o}\mathsf{T}$ as

$$ {}^{o}P' = \mathsf{R}(\dot{\psi}\Delta t)\,({}^{o}P - {}^{o}P_{\mathrm{rot}}) + {}^{o}P_{\mathrm{rot}} + {}^{o}\mathsf{T} \tag{2} $$

where Δt is the time interval between the current and the previous discrete time instance, R is a 3×3 rotation matrix around the height axis by angle $\dot{\psi}\Delta t$, and

$$ {}^{o}\mathsf{T} = \begin{bmatrix} \frac{v+a\Delta t}{\dot{\psi}} \left(1 - \cos(\dot{\psi}\Delta t)\right) \\ 0 \\ \frac{v+a\Delta t}{\dot{\psi}} \sin(\dot{\psi}\Delta t) \end{bmatrix}. \tag{3} $$

Applying (2) to the reference point ${}^{o}P_{\mathrm{ref}} = [0, 0, 0]^{\mathsf{T}}$ yields the predicted position of the vehicle in object coordinates ${}^{o}P'_{\mathrm{ref}}$. Since the reference point in the state vector is defined in ego coordinates, this position has to be transformed from object to ego coordinates by rotating around the negative ψ and translating the result by $[{}^{e}X_{\mathrm{ref}}, 0, {}^{e}Z_{\mathrm{ref}}]^{\mathsf{T}}$.

The following time-discrete system equations define the nonlinear system model:

$$ \Delta\boldsymbol{x} = \begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta X_{\mathrm{rot}} \\ \Delta Z_{\mathrm{rot}} \\ \Delta\psi \\ \Delta\dot{\psi} \\ \Delta v \\ \Delta a \end{bmatrix} = \begin{bmatrix} \mathsf{R}_1^{-1}(\psi)\,{}^{o}P'_{\mathrm{ref}} \\ \mathsf{R}_3^{-1}(\psi)\,{}^{o}P'_{\mathrm{ref}} \\ 0 \\ 0 \\ \dot{\psi}\Delta t \\ 0 \\ a\Delta t \\ 0 \end{bmatrix}. \tag{4} $$

Finally, the predicted state vector $\widehat{\boldsymbol{x}}^{-}$ at time $t + \Delta t$ is computed as

$$ \widehat{\boldsymbol{x}}^{-}(t + \Delta t) = \widehat{\boldsymbol{x}}(t) + \Delta\boldsymbol{x}. \tag{5} $$

3) Measurement Model: From the perspective of the stereo vision sensor, a given object point ${}^{o}P_m$ with $0 \le m < M$ can be observed in terms of a subpixel accurate image position $p_m = (u_m, v_m)$ and stereo disparity d_m at this position. We will denote $\boldsymbol{z_m}(t) = [u_m(t), v_m(t), d_m(t)]^{\mathsf{T}}$ as the measurement of point ${}^{o}P_m$ in the image at time t and ${}^{o}\tilde{P}_m(t)$ as the corresponding measurement in object coordinates. In the following, the time index is skipped for better readability when all variables refer to the same time instance.

If the association between an object point and an image position is known at one point in time, then it is possible to reassign it by tracking the image position over time using a feature tracker, such as the well-known Kanade–Lucas–Tomasi tracker [38].

All M measurements build the total measurement vector $\boldsymbol{z} = [z_0^{\mathsf{T}}, \ldots, z_{M-1}^{\mathsf{T}}]^{\mathsf{T}}$. The measurement equations defining the nonlinear relation $\widehat{\boldsymbol{z}} = h(\widehat{\boldsymbol{x}}^{-})$ result from a perspective camera model "

$$ \widehat{u}_m = h_{m,1}(\widehat{\boldsymbol{x}}^{-}) = f_u \frac{{}^{c}X_m}{{}^{c}Z_m} + u_0 \tag{6} $$

$$ \widehat{v}_m = h_{m,2}(\widehat{\boldsymbol{x}}^{-}) = -f_v \frac{{}^{c}Y_m}{{}^{c}Z_m} + v_0 \tag{7} $$

$$ \widehat{d}_m = h_{m,3}(\widehat{\boldsymbol{x}}^{-}) = f_u \frac{b}{{}^{c}Z_m} \tag{8} $$

where f_u and f_v are the horizontal and vertical focal lengths of the camera, (u_0, v_0) is the principal point, and b is the baseline of the stereo system. ${}^{c}P_m = [{}^{c}X_m, {}^{c}Y_m, {}^{c}Z_m]^{\mathsf{T}}$ is the point ${}^{o}P_m$ in camera coordinates. The total transformation consists of an object to ego transformation (parameterized by the state variables ψ, ${}^{e}X_{\mathrm{ref}}$, and ${}^{e}Z_{\mathrm{ref}}$) and the constant transformation between ego and camera system (including the extrinsic camera parameters).

The 3×3 noise matrix for a given point measurement is assumed to be constant and uncorrelated. It is defined as $\Gamma_m = \mathrm{diag}(\sigma_u^2, \sigma_v^2, \sigma_d^2)$. The resulting total measurement noise matrix Γ of size $3M \times 3M$ is the block diagonal matrix $\Gamma = \mathrm{blkdiag}(\Gamma_0, \Gamma_1, \ldots, \Gamma_{M-1})$.

Following the extended Kalman filter approach, h has to be linearized at the current estimated state $\widehat{\boldsymbol{x}}^{-}$ for the computation of the Kalman gain and the propagation of the state covariance matrix, which yields

$$ H = \left[\begin{matrix} \frac{\partial h_{0,1}}{\partial \boldsymbol{x}} \\ \frac{\partial h_{0,2}}{\partial \boldsymbol{x}} \\ \frac{\partial h_{0,3}}{\partial \boldsymbol{x}} \\ \vdots \\ \frac{\partial h_{M-1,1}}{\partial \boldsymbol{x}} \\ \frac{\partial h_{M-1,2}}{\partial \boldsymbol{x}} \\ \frac{\partial h_{M-1,3}}{\partial \boldsymbol{x}} \end{matrix} \right]_{\boldsymbol{x}=\widehat{\boldsymbol{x}}^{-}}. \tag{9} $$

The tracking performance increases with the number of contributing points. A varying number of M points due to loss

or addition of newly tracked points can easily be handled, since each point simply adds another three measurements to the filter.

C. Kalman Filter Initialization

Before object tracking can be started, the filter has to be initialized with the pose and motion parameters of an object hypothesis. Two different initialization methods have been realized: one is based on computer vision only, whereas the other fuses radar information with image data.

1) Image-Based Initialization: A method called *6D-Vision* [33] is used as input for object detection. The main idea of *6D-Vision* is to track image features with depth known from stereo over a sequence of consecutive frames. This yields a 3-D position for each point at a given time step, fusing the information of space and time. A Kalman filter is used for the stabilization of noisy observations. At the same time, it allows for estimating the 3-D point motion.

The filter state vector contains the 3-D position of a point plus a 3-D velocity component; thus, it is also referred to as *6D vector* in the following.

Fig. 3 shows an example *6D-Vision* result taken from a sequence of images with four moving objects. The arrows point to the position where a given point (projected onto the image plane) will be in half a second based on the current filter state. The superimposed white dots indicate stationary points in the scene. Given these data, humans easily group the arrows belonging to (1) the pedestrian with the stroller and (2) the car turning into the street from the right. Having a closer look, even the (3) oncoming car, slowly approaching, can be detected, as well as (4) a vehicle moving from left to right at an intersection ahead.

A group of 6-D vectors within a local neighborhood with compatible motion indicates an object candidate. Now, the task is to automate the object detection using clustering algorithms. In our system, a fast histogram-based clustering is used, where *compatibility* is defined in terms of the Mahalanobis distance. A cluster with sufficient support (e.g., number of points above a threshold) generates a new object hypothesis.

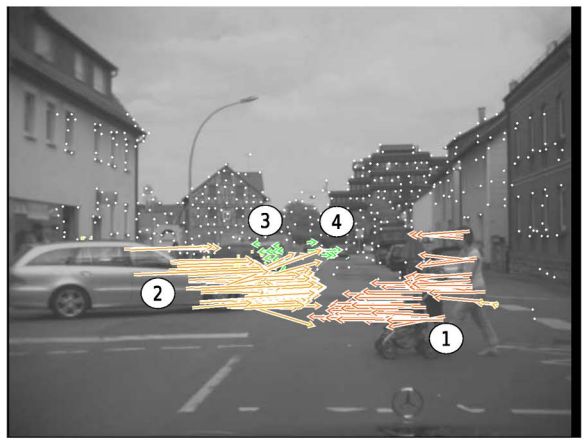
The centroid of the 3-D positions ${}^eP_C = [{}^eP_{C_x}, {}^eP_{C_y}, {}^eP_{C_z}]^\mathsf{T}$ in ego coordinates and the mean velocity vector $\bar{V} = [\bar{v}_x, \bar{v}_y, \bar{v}_z]^\mathsf{T}$ of the corresponding point cloud define the initial object pose and velocity (assuming the average moving direction of all vectors gives us the object orientation). This leads to the following filter state $\widehat{\boldsymbol{x}}$ at initialization time t_0:

$$\widehat{\boldsymbol{x}}(t_0) = \left[{}^eP_{C_x}, {}^eP_{C_z}, 0, 0, \psi_0, \|\bar{V}\|, 0, 0\right]^\mathsf{T} \tag{10}$$

where $\psi_0 = \arccos(\bar{v}_z \|\bar{V}\|^{-1})$. It is possible to restrict the initialization method to objects exceeding a certain velocity threshold, motion direction, or dimensional constraint.

2) Radar-Based Initialization: One drawback of the image-based initialization method previously described is the delay between initialization of 6-D vectors and generating an object hypothesis. It takes a few time steps until the Kalman-filtered point tracks are reliable for clustering, particularly at large distances due to stereo uncertainty, which quadratically increases with distance. Depending on the relative velocity and cycle time, an oncoming vehicle can easily approach up to 30 m in this time period. Since the Kalman filter estimating the object's motion state also requires a couple of cycles to converge, the objective should be to initialize object tracking as early as possible.

(a)



(b)

Fig. 3. *6D-Vision* results of a scene with four moving objects. (1) Pedestrian with a stroller walking to the left. (2) Vehicle turning into the street from the right. (3) Slow oncoming vehicle. (4) Vehicle moving from left to right at a far distance. The arrows indicate the predicted linear point motion for the next half second. Clustering of neighboring 6-D vectors with common motion yields object candidates. (a) Example of an image sequence. (b) Results superimposed.

Radar sensors allow for detecting oncoming vehicles at distances of up to 200 m. For comparison, the maximum reliable distance of a stereo system with 0.3-m baseline, as used in our demonstrator car, is about 50–60 m. The radar is used to initialize the filter state as follows:

$$\widehat{\boldsymbol{x}}(t_0) = [{}^eP_{\mathrm{radar}_x}, {}^eP_{\mathrm{radar}_z}, 0, 0, \psi_0, v_{\mathrm{radar}}, 0, 0]^\mathsf{T} \tag{11}$$

where ${}^eP_{\mathrm{radar}_x}$ and ${}^eP_{\mathrm{radar}_z}$ represent the lateral and longitudinal positions of the radar target in ego coordinates, and $v_{\mathrm{radar}} = v_{\mathrm{rel}} + v_{\mathrm{ego}}$ is the absolute radar velocity of the object. The initial orientation ψ_0 can be derived from a number of previous radar positions using linear regression [39]. Initializing the tracker with $\psi_0 = \pi$, i.e., the reversed heading direction of the ego-vehicle, is a fast alternative in practice.

D. Point Model Initialization

So far, we have assumed that the complete 3-D point model is known *a priori*. However, in practice, the object's shape is unknown and has to be obtained from the (noisy) observations.

Using the image-based initialization method, the initial 3-D object point model is directly available from the 3-D point positions of the 6-D vectors building the object hypothesis.

If initialized by radar, then we have to find image features corresponding to points compatible with the object in location and motion. The radar target is used to generate a candidate image region likely to contain the vehicle to track. Since distance, velocity, and rough orientation of the object are known from the radar, it is possible to predict image displacements and disparity values for features within this image region. The 3-D points corresponding to the image features that match the expectation are added to the object model.

In both cases, the initial Kalman filter state defines the local object coordinate system to which all points are referred.

E. Point Model Update

The initial object model can be quite noisy and incomplete and will usually contain outliers [see Fig. 4(a)]. Let ${}^{O}\mathcal{P}_m$ denote the true 3-D position of a concrete point instance on the surface of object O with respect to the ideal object coordinate system. The actual error between ${}^{O}\mathcal{P}_m$ and the estimated object model point ${}^{o}P_m$ consists of the measurement uncertainty of the stereo system and the deviation between the ideal and the estimated object coordinate system (including object pose parameters).

The optimal solution minimizing this error would be the integration of all point positions into the filter state, leading to one large state vector. Then, the Kalman filter would estimate the object's structure together with the pose and motion parameters. However, with an expected number of 50–300 points for one object, the larger filter state significantly increases the overall computation time and may lead to numerical problems if the number of points gets too large.

With respect to real-time demands, we separate the problem of motion estimation from the problem of shape reconstruction. One can show that for each object point ${}^{o}P_m$ observed several times in terms of ${}^{o}\tilde{P}_m(t)$, a maximum-likelihood estimation is given by

$$ {}^{o}P_m(t+1) = \left(\sum_{j=t_m}^{t} C_m^{-1}(j) \right)^{-1} \sum_{j=t_m}^{t} C_m^{-1}(j)\, {}^{o}\tilde{P}_m(j) \qquad (12) $$

under the assumption of uncorrelated measurements and zero-mean Gaussian measurement noise [40]. $C_m(t)$ denotes the 3×3 covariance matrix of ${}^{o}\tilde{P}_m(t)$, and t_m denotes the time step point ${}^{o}P_m$ that has been added to the model.

If the uncertainties are ignored, i.e., $C_m(t) = I \ \forall t$, (12) reduces to the simple averaging of the point measurements, i.e.,

$$ {}^{o}P_m(t+1) = \frac{(t - t_m)\, {}^{o}P_m(t) + {}^{o}\tilde{P}_m(t+1)}{t - t_m + 1} \qquad (13) $$

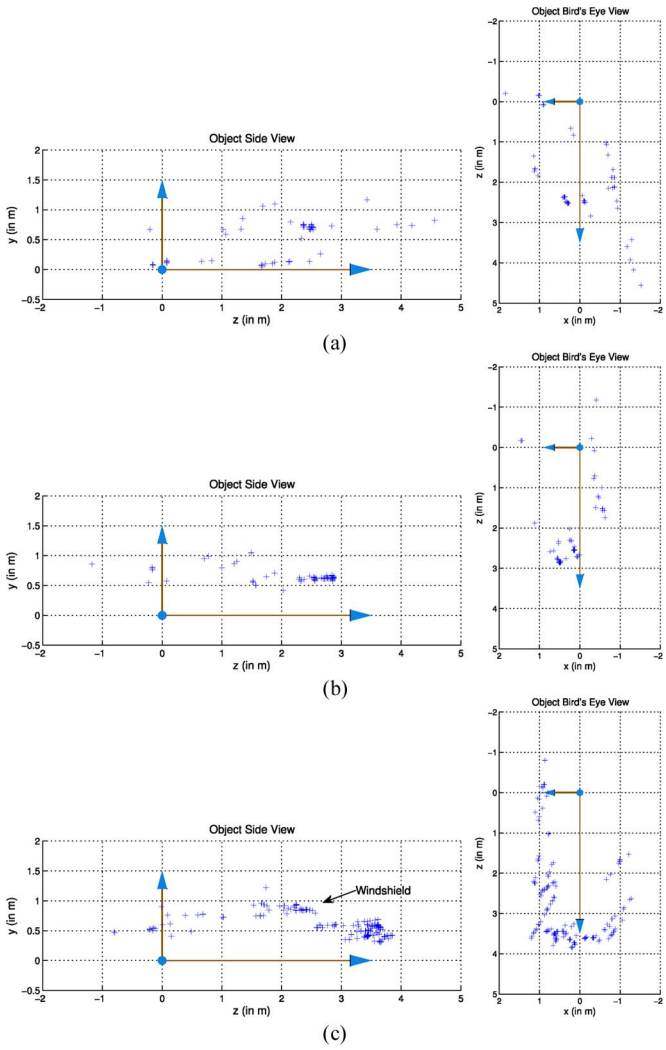


Fig. 4. Side view and bird's-eye view of the 3-D point model of an oncoming vehicle at different time steps. The initial noisy point cloud is refined over time, and the shape of a vehicle becomes visible. All points are given in object coordinates. See Fig. 12 for corresponding images. (a) Frame 57 (initialization of model). (b) Frame 75. (c) Frame 130.

which can be computed very fast. It should be noted that the later equation has been used for updating points in all of our experiments due to real-time constraints.

Beside refining the existing model, adding new points to the model at runtime is very important, since image feature tracks can get lost, for example, due to occlusions or image brightness inconsistency. To compensate for lost point tracks, in each cycle, new feature tracks up to a user-defined maximum number are initialized in the image and assigned to an object if they are compatible in position and velocity.

On the other hand, outliers in the object model are detected and removed when the residual differences between measured and predicted values are outside a given confidence interval.

IV. Experimental Results

We will first evaluate the performance of the proposed method on synthetic data with the available ground truth. Then, real-world results will be shown, addressing different scenarios and aspects of vehicle tracking.

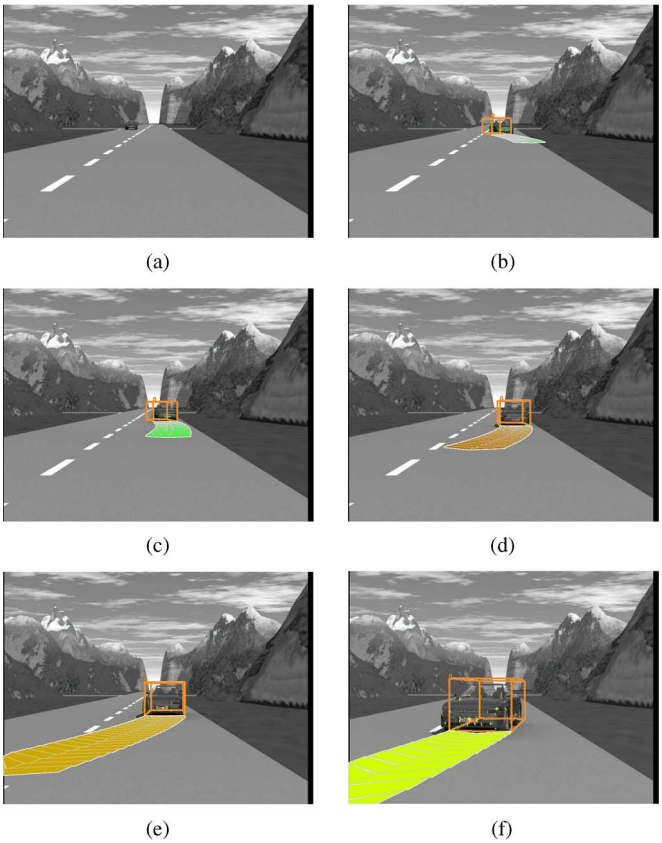


Fig. 5. Selected frames of a synthetic sequence taken at different time steps. The object's bounding box and the predicted motion path for the next second, estimated by the Kalman filter, are visualized. The marked points correspond to the tracked image features used for initialization and object tracking. (a) Frame 0. (b) Frame 40. (c) Frame 50. (d) Frame 60. (e) Frame 70. (f) Frame 80.

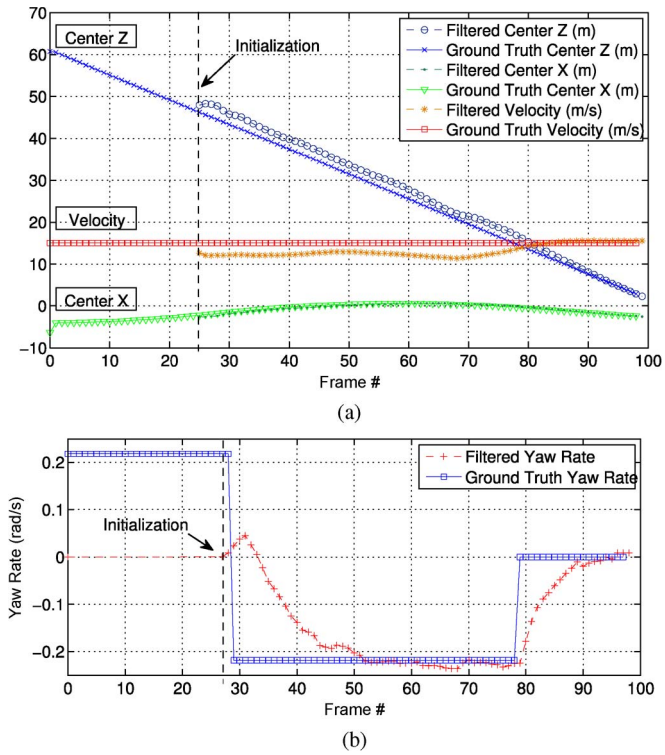


Fig. 6. Filtering results of the virtual sequence compared with ground truth data. (a) Position + velocity. (b) Yaw rate.

A. Simulation Results

In Fig. 5, selected frames of a rendered scene are shown. The scene contains a vehicle driving toward the stationary ego-vehicle (starting at 60-m distance on the opposite lane with a constant velocity of 15 m/s). After 30 frames, the observed vehicle changes the lane to simulate a potentially critical situation and turns back to the opposite lane just in front of the ego-vehicle. The constant time between two frames is 0.04 s. The 3-D bounding box, which is highlighted in the foregoing figures, indicates position, dimension, and orientation. The predicted driving path for the next second is visualized as carpet on the ground plane. Its curvature is related to the yaw rate, whereas the length of the path is based on the vehicle's estimated velocity and acceleration.

The resulting state estimates for position (Center$X$, Center$Z$) and velocity compared with ground truth data are shown in Fig. 6(a). The vehicle is detected at approximately 50-m distance at frame 25 and initialized with a velocity of 12 m/s. The plot shows an almost ideal reconstruction of the lateral position, whereas the longitudinal distance is a bit larger in the filtered results due to the underestimated velocity up to frame 82. The *root-mean square error* (RMSE) of the lateral position is 0.2728 m over the total sequence and reduces to 0.1287 m when only the results after frame 80 are considered. Accordingly, the RMSE of the longitudinal distance is 2.0044 m over the total sequence and 0.8565 m for the results after frame 80, respectively. The underestimated initial velocity leads to a total RMSE in the velocity of 2.2538 and 0.4934 m/s after frame 80.

Fig. 6(b) visualizes how the filter reacts to sudden (unnatural) changes of the yaw rate. As can be seen, these changes cannot directly be adapted by the filter, leading to larger deviations from the ground truth and a total RMSE of 0.0980 rad/s. However, the smooth adaptation corresponds to the expected characteristic filter behavior. After convergence, the ground truth value is well approximated. The results also show that the filter converges faster as the vehicle comes closer due to the lower uncertainty of the measurements.

B. Real-World Results

The proposed system has been integrated into our Mercedes-Benz S-class demonstrator car *UTA* for extensive real-world experimentation and testing.

1) Country Road Curve I: As mentioned in Section I, collision-avoidance systems must not produce false alarms if an oncoming vehicle approaches on a left curve.

Fig. 7 shows selected frames of a sequence with an oncoming vehicle, i.e., a Mercedes-Benz B-class with no preparations, on a curved country road. It illustrates how the Kalman filter sequentially corrects the inaccurate initialization within the first 500 ms of tracking. The overlaid carpets represent the predicted driving corridor of the ego-vehicle based on inertial sensors and the estimated driving path of the oncoming vehicle,

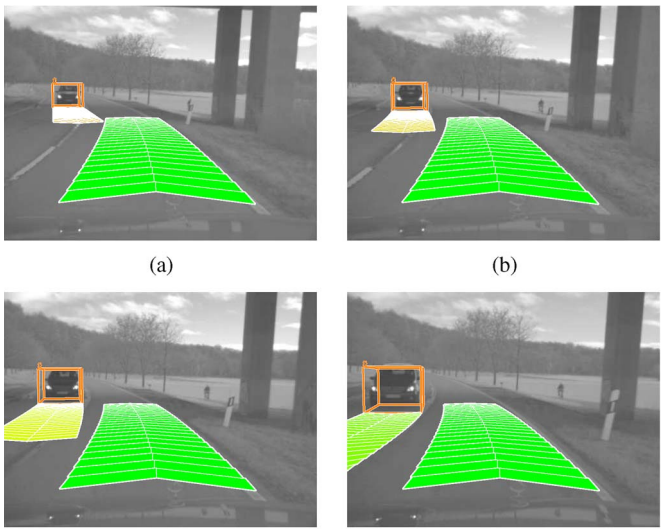


Fig. 7. Selected frames of a real-world scene with one oncoming vehicle captured in the left curve ($v_{\mathrm{rel}} \approx 30$ m/s). The driving corridor of the ego-vehicle based on inertial sensors is visualized in all images. It can be observed how the filter successively corrects the erroneous initial assumption of the driving path. (a) Frame 190. (b) Frame 195. (c) Frame 200. (d) Frame 205.

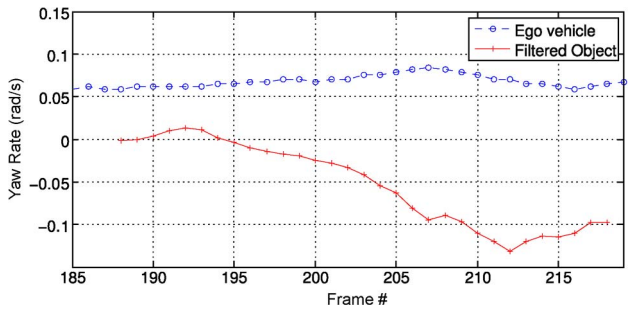


Fig. 8. Filtering results of the country road sequence. The filtered yaw rate indicates that the oncoming vehicle is steering to the right, whereas the yaw rate of the ego-vehicle is consistent with a slight left turn.

respectively. At frame 188, the oncoming vehicle is detected at about 38-m distance using the purely image-based method, as introduced in Section III-C. The relative velocity v_{rel} is about 30 m/s.

Tracking the object over a few frames indicates that the object is not driving straight ahead (critical) but, rather, on a curved path (uncritical). With the assumed motion model, the motion of all observed points on the object can best be explained as a vehicle with a negative yaw rate. The estimated yaw rate of the oncoming vehicle is plotted in Fig. 8. It can be seen that the absolute value of the estimated yaw rate of the oncoming vehicle is slightly larger than that of the ego-vehicle, since the oncoming vehicle is driving on the inner lane of the curve on a smaller radius. It is important to state that the system does not include any lane recognition component. The estimated yaw rate purely depends on the point motion and the underlying motion model.

Critically reflecting the results, it can be observed that there is not much more than a second left between object detection and passing. The filter requires 10–15 cycles (0.4–0.6 s) until the driving path is well estimated. In this experiment, the speed of

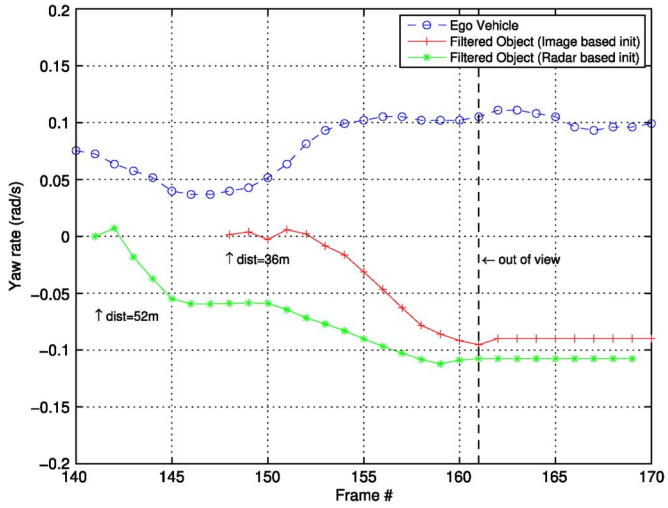


Fig. 9. Effect of the different initialization methods on the yaw rate estimate for the country road curve scenario with $v_{\mathrm{rel}} \approx 44$ m/s.

both vehicles is approximately 50–60 km/h. The speed limit on this road is 70 km/h. In practice, particularly in potentially critical situations, one can expect even higher relative velocities.

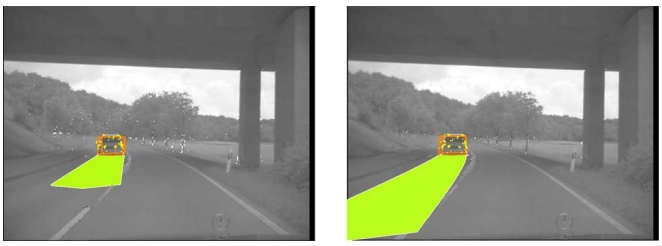
Since the Kalman filter always requires a couple of cycles until convergence, initializing the filter as early as possible becomes even more important.

2) Country Road Curve II: We have repeated the country road curve experiment, as explained in case 1, with a relative velocity of 44 m/s, i.e., each vehicle is driving at approximately 80 km/h. This time, the ego-vehicle has been equipped with a 77-GHz radar able to detect oncoming moving objects at distances of up to 200 m within a 18° field of view.

With the image-based method, this time, the object is detected at 36-m distance. Due to the relative velocity, only 13 cycles remain until the vehicle leaves the field of view of the cameras. The radar detects the object at a 52-m distance, which is detected 16 m earlier compared with the image-based method, and this corresponds to seven additional image pairs to evaluate. In this scene, the object is not in the field of view of the radar earlier due to the curve. On straight roads, one can expect oncoming objects to be detected at significantly larger distance by the radar.

The estimated yaw rates are shown in Fig. 9. Both runs have been initialized with $\dot{\psi} = 0.0$ rad/s and converge toward approximately -0.1 rad/s. However, the yaw rate of the filter initialized by radar clearly indicates that the object is driving on a circular path, whereas the filter initialized using the image-based method is still indicating linear motion. Fig. 10 visualizes the estimated motion state at frame 151, i.e., shortly after the object has been detected by the image-based method.

In addition to starting the tracking earlier, initializing the object's motion state with the radar velocity also improves the result, as can be seen in Fig. 11. Using the image-based initialization method, the initial velocity is underestimated. It takes about five cycles until the real object velocity is accurately estimated by the filter. A good estimate of the object's velocity also provides a strong cue for separating points on the object's surface from static scene points.



(a) (b)

Fig. 10. Frame 151 of the country road scene with the predicted driving path superimposed. Due to earlier object detection, the yaw rate estimate of the filter initialized by radar already indicates a curved path as the image-only filter still assumes linear motion. The velocity in (a) is also underestimated. (a) Initialized from image. (b) Initialized from radar.

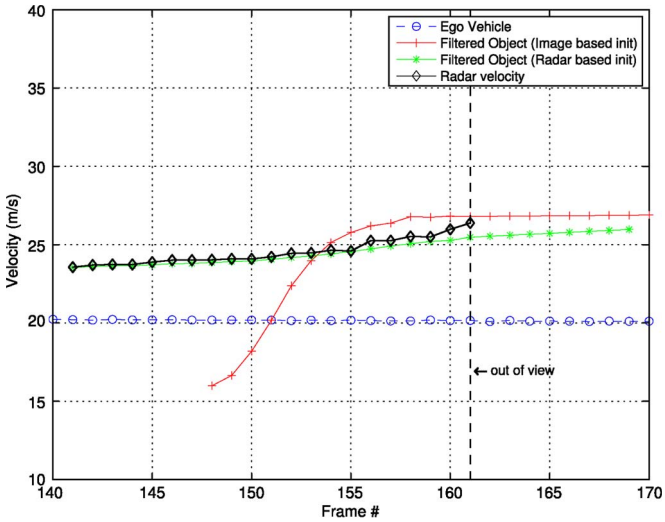


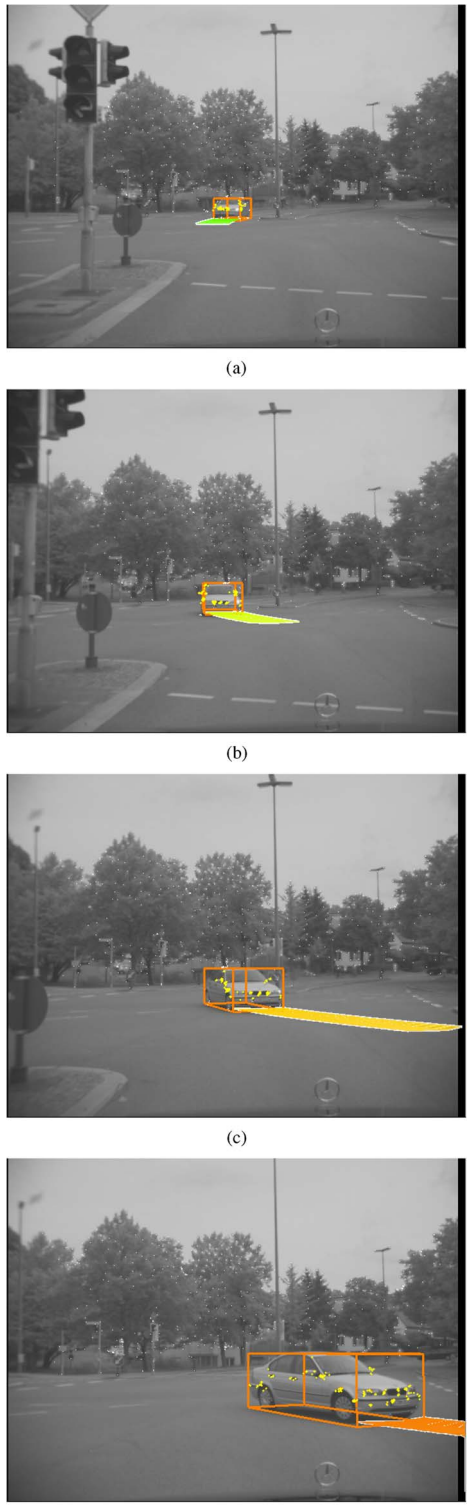
Fig. 11. Initializing the object's motion state with the radar velocity significantly improves the velocity estimate.

3) Oncoming Traffic at Intersections: Fig. 12 shows an urban intersection scenario. Both the ego-vehicle and the oncoming vehicle turn left. Similar to the previous results, the image overlay shows the filtered object state in terms of pose (bounding box), object model (marked features on the object surface), and motion state (carpet on the ground indicating the predicted driving path for the next second). The resulting object trajectories are shown in Fig. 13.

Let us assume that the driver in the ego-vehicle wants to drive straight ahead in the foregoing example scene. A free-space analysis based on static occupancy grids, for example, [41], would result in an obstacle-free driving corridor for the ego-vehicle up to frame 125 [see Fig. 12(a)–(c)]. Obviously, ignoring there is another vehicle at the intersection that will be crossing the ego-vehicle's driving path can lead to potentially critical situations. Thus, the proposed system provides valuable information for predicting collisions.

4) Leading Vehicles: The proposed method is not restricted to oncoming traffic. Without adaptation, it can be used to track vehicles driving in any arbitrary direction. Fig. 14 shows some examples of the tracking results of leading vehicles.

In many cases, tracking leading vehicles is even less difficult compared with oncoming traffic due to the lower relative velocities. Thus, there is more time for filtering. Furthermore, leading vehicles are usually detected at closer distances, i.e., the 3-D position of points can more precisely be extracted due to lower stereo uncertainty, which directly improves the initial object model.



(a) (b) (c) (d)

Fig. 12. Filtering result of a left-turning oncoming vehicle at an urban intersection. The superimposed crosses mark the feature positions used in building the object model. (a) Frame 75. (b) Frame 100. (c) Frame 125. (d) Frame 130.

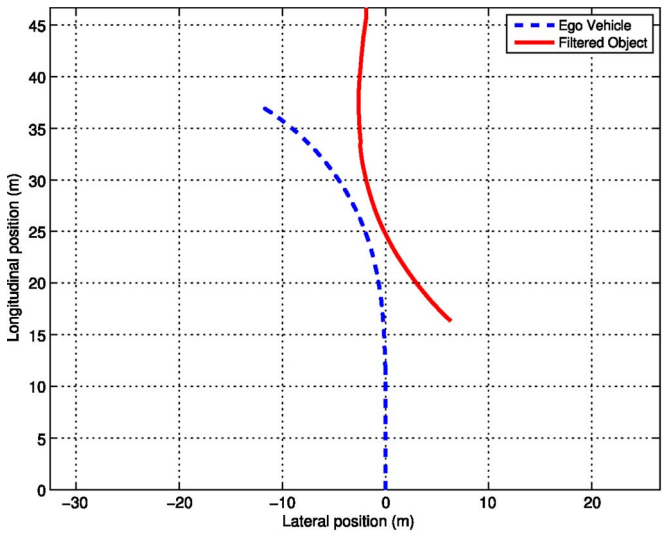


Fig. 13. Resulting trajectories of ego-vehicle and filtered object for the urban intersection scene.

5) Partial Occlusions: Modeling an object as 3-D point cloud is beneficial if partial occlusions occur, as can be seen in Fig. 14(d). In this example, the pedestrian with stroller is crossing the street in front of the ego-vehicle, whereas a vehicle turns into the street behind the pedestrian from the left (this scene has already been addressed in Fig. 3). Due to a minimum velocity threshold, an object hypothesis is only created for object (2).

The object has been detected before partial occlusion. During each time step, the system tries to add new points with compatible distance and motion to the object model. These new points are registered to the current object model and compensate for the lost feature points that are currently occluded. If sufficient traceable points on the object's surface are present, then the object motion state can accurately be estimated, independent of partial occlusions. In this example, the incompatible motion and distance of feature points found on the pedestrian prevents wrongly adding these points to the object model. Model knowledge of the object's dimension from earlier time steps without occlusion helps to find new points on the car's front, although they are separated from the majority of points at the car's rear and prevents the object's model from being split into two parts.

6) Challenges and Limits: Since the system is intended to reliably run in a real-world environment, it has to be robust to cluttered and noisy data.

Fig. 15 shows an oncoming vehicle detected and tracked in a rainy night. The vehicle moves toward the ego-vehicle and suddenly turns left to drive around the obstacle. Only the headlights and the number plate illuminated by the ego-vehicle are visible. This information, however, is sufficient to accurately predict the driving path of the vehicle, providing the capabilities of the proposed method.

However, this system also has limits. Fig. 16 gives two examples of where the system fails to detect or track an object. In Fig. 16(a), the contrast on the object's surface and between the object and the surrounding background is very poor due to the extreme backlight. The system fails to find sufficient features to track on the moving objects in the scene. In addition,

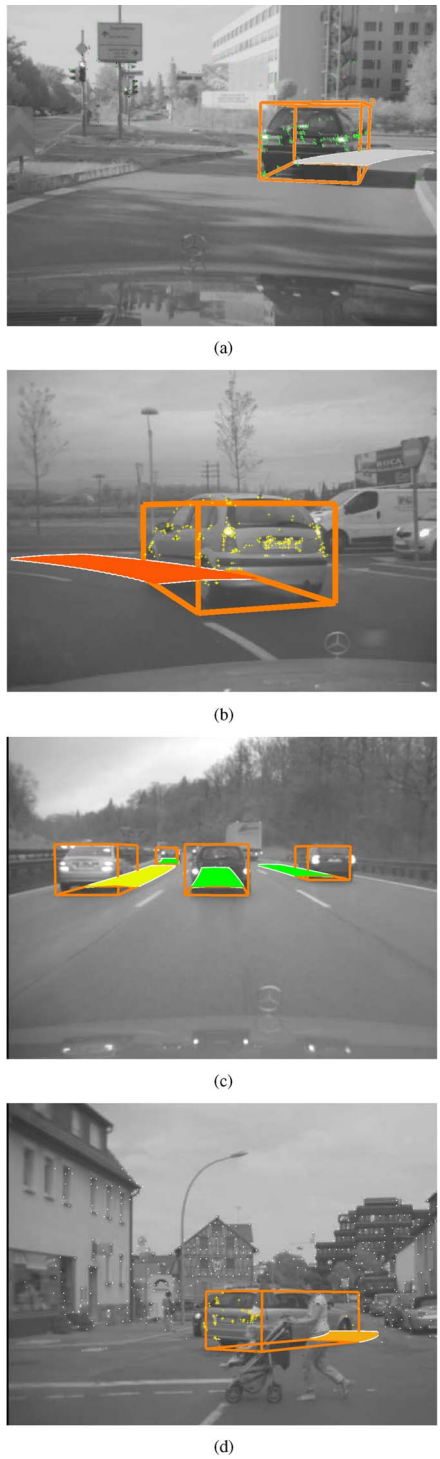


Fig. 14. Proposed method is not limited to oncoming traffic and single-object tracking. The examples show different tracking results of leading vehicles. Due to the variable object model, partial occlusions as in (d) do not disturb the tracking, as long as enough points are present to create the object model. (a) Right turning. (b) Traffic circle. (c) Highway. (d) Partly occluded left turning.

strong reflections with different influence on the left and right images impair the depth computation ability from stereo. In Fig. 16(b), an object has been detected a few frames earlier in this extreme rainy scene. Due to the considerable accumulation of water on the windshield, the images get blurred, and again, the contrast is very poor. In such scenes, the image brightness

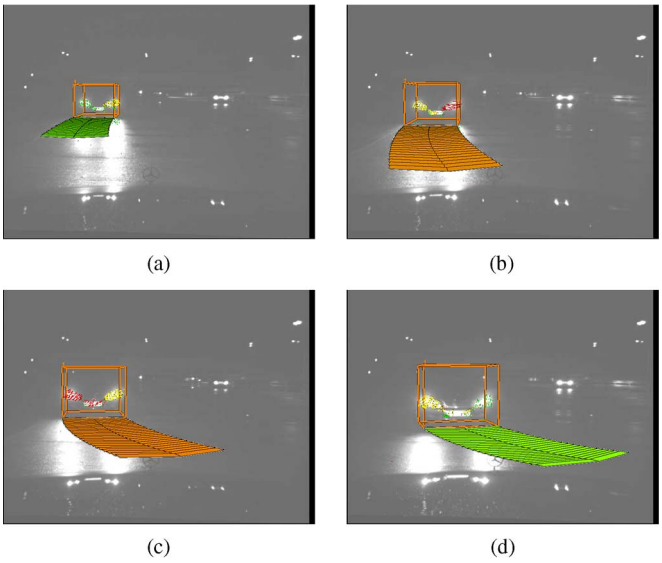


Fig. 15. Tracking results of a rainy night scene. Although only points on the headlights can be tracked, the driving path is accurately estimated. (a) Frame 150. (b) Frame 160. (c) Frame 170. (d) Frame 180.

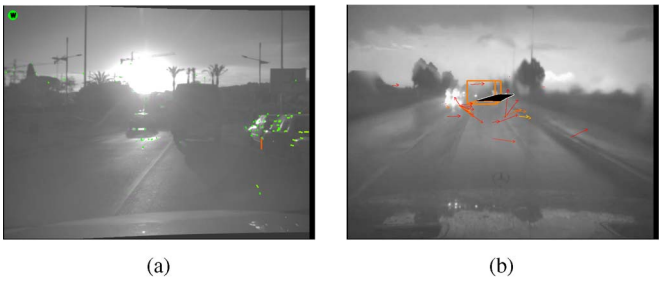


Fig. 16. Object tracking fails if reliable feature tracking on an object's surface is not possible, e.g., if the brightness constancy assumption does not hold or contrast in the image is not sufficient. However, in most situations in which the system fails, humans also have difficulties interpreting the scene accurately. (a) Extreme back light. (b) Extreme rain.

constancy assumption, which is used for feature tracking, usually does not hold. The obtained few flow vectors (ego-motion compensated), as shown superimposed in the image, are error prone and contain no valuable information for object tracking. However, in most situations that the system fails, humans also have difficulties accurately interpreting the scene.

V. Conclusion

We have proposed a new method for the image-based real-time tracking of vehicles from a moving platform.

Results of experiments with synthetic data have proven that the system is capable of accurately estimating the motion and pose parameters of an oncoming vehicle, including the object's yaw rate. Results from real-world experiments have addressed oncoming traffic scenarios on country roads and at urban intersections. However, the system's capabilities are not restricted to oncoming traffic and can be extended to tracking a leading vehicle using the same method without any additional adaptations.

Two different object detection methods have been presented. The first method is purely image based and initializes objects based on clusters of Kalman-filtered 3-D point tracks with common motion state. The second method fuses vision with radar. The radar provides a good initial estimate on velocity and distance. This also helps to find features with image displacements compatible in velocity, moving direction, and distance that can be assigned to the object model.

The combined initialization method significantly improves the motion state estimates for country road traffic scenarios, since objects are detected earlier and tracking can be started without delay. With this method, the correct driving path of an oncoming vehicle is predicted purely based on the object's estimated motion.

The feature-based object point model does not require *a priori* knowledge about the object's shape. Instead, the object model is extracted and refined from the observations over time. The advantage of this object model is that it does not have to be complete. The rainy night scene example has shown that very accurate tracking results can be achieved, even if only the headlights and the illuminated number plate of the oncoming vehicle are visible. Furthermore, the object model is robust to partial occlusions.

Continuously adding new points to compensate for the loss of tracked features increases the robustness of the system. The more points are contributing to building the model, then the less sensitive the system becomes to outliers. The underlying dynamic model represents a constraining force by restricting lateral movements to circular path motion.

On current PC hardware (Intel Core 2 Quad), the proposed system requires a total processing time of less than 40 ms for 640×480 images, i.e., runs with a stable frame rate of 25 Hz, without exploiting all the capacities of parallel processing or computation on the system's graphics card. The object tracking itself requires less than 2 ms of processing time for a single object.

The proposed method enables a variety of potential applications. For example, it can be used as input for future collision-avoidance systems or to improve the free-space prediction in front of the ego-vehicle.

However, predicting the driving path of oncoming vehicles based on vision remains a challenging field of research. In general, driving maneuvers can have a wide dynamic range, depending on various traffic situations, reaching from quick turn maneuvers in inner cities to quite smooth driving on highways. A single Kalman filter parameterization can hardly cover all these different dynamic situations equally well. The filter should be smooth in low dynamic situations but be able to follow, even in high dynamic situations. One solution to this problem could be a multifilter approach. A number of Kalman filters parameterized with variances for different scenarios (e.g., linear movement, curve movement, or turn maneuver) could be run in parallel with an automated selection of the currently best-matching model.

Another challenging field for further improvements is the applicability of the approach in dense traffic scenes. Assigning feature points to objects becomes a nontrivial segmentation problem when more than one object is moving with equal velocity side by side or bumper to bumper.

REFERENCES

[1] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 1068–1073.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. New York: Wiley, 2001.

[3] D. Simon, *Optimal State Estimation*. New York: Wiley, 2006.

[4] D. Koller, K. Daniilidis, and H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, Jun. 1993.

[5] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," in *Proc. ECCV*, 1994, pp. 189–196.

[6] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proc. Comput. Vis. Pattern Recog.*, San Juan, Puerto Rico, 1997, pp. 495–501.

[7] S.-C. Chen, M.-L. Shyu, and C. Zhang, "An intelligent framework for spatio-temporal vehicle tracking," in *Proc. IEEE Intell. Transp. Syst.*, Aug. 25–29, 2001, pp. 213–218.

[8] Z. Kim and J. Malik, "Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking," in *Proc. ICCV*, 2003, pp. 524–531.

[9] L. Jianguang, T. Tieniu, H. Weiming, Y. Hao, and S. Maybank, "3-D model-based vehicle tracking," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1561–1569, Oct. 2005.

[10] N. Kanhere and S. Birchfield, "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 148–160, Mar. 2008.

[11] H. Badino, "A robust approach for ego-motion estimation using a mobile stereo platform," in *Proc. 1st IWCM*, Guenzburg, Germany, Oct. 12–14, 2004, pp. 198–208.

[12] J. Klappstein, "Optical-flow based detection of moving objects in traffic scenes," Ph.D. dissertation, Univ. Heidelberg, Heidelberg, Germany, 2008.

[13] M. Brauckmann, C. Goerick, J. Gross, and T. Zielke, "Towards all around automatic visual obstacle sensing for cars," in *Proc. IEEE Intell. Veh. Symp.*, Paris, France, Oct. 1994, pp. 79–84.

[14] S. Smith and J. Brady, "ASSET-2: Real-time motion segmentation and shape tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 814–820, Aug. 1995.

[15] F. Dellaert and C. Thorpe, "Robust car tracking using Kalman filtering and Bayesian templates," in *Proc. Conf. Intell. Transp. Syst.*, 1997, pp. 72–83.

[16] K. She, G. Bebis, H. Gu, and R. Miller, "Vehicle tracking using on-line fusion of color and shape features," in *Proc. 7th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 3–6, 2004, pp. 731–736.

[17] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair, and A. Broggi, "Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2364–2375, Aug. 2006.

[18] C. Fu, C. Huang, and Y. Chen, "Vision-based preceding vehicle detection and tracking," in *Proc. Int. Conf. Pattern Recog.*, 2006, pp. II:1070–II:1073.

[19] H. Liu, F. Sun, and K. He, "Symmetry-aided particle filter for vehicle tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 10–14, 2007, pp. 4633–4638.

[20] H. Liu, F. Sun, L. Yu, and K. He, "Vehicle tracking using stochastic fusion-based particle filter," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Oct. 29–Nov. 2, 2007, pp. 2735–2740.

[21] M. Maehlisch, W. Ritter, and K. Dietmayer, "De-cluttering with integrated probabilistic data association for multisensor multitarget ACC vehicle tracking," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 13–15, 2007, pp. 178–183.

[22] Y. Chen, M. Das, and D. Bajpai, "Vehicle tracking and distance estimation based on multiple image features," in *Proc. 4th Can. Conf. Comput. Robot Vis.*, 2007, pp. 371–378.

[23] N. Kaempchen, M. Buehler, and K. Dietmayer, "Feature-level fusion for free-form object tracking using laserscanner and video," in *Proc. IEEE Intell. Veh. Symp.*, Las Vegas, NV, Jun. 2005, pp. 453–458.

[24] H. Cheng, N. Zheng, X. Zhang, J. Qin, and H. van de Wetering, "Interactive road situation analysis for driver assistance and safety warning systems: Framework and algorithms," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 157–167, Mar. 2007.

[25] W. van der Mark and D. M. Gavrila, "Real-time dense stereo for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 38–50, Mar. 2006.

[26] U. Franke and I. Kutzbach, "Fast stereo based object detection for stop and go traffic," in *Proc. IEEE Intell. Veh. Symp.*, Tokyo, Japan, Sep. 1996, pp. 339–344.

[27] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through 'v-disparity' representation," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2002, vol. 2, pp. 646–651.

[28] U. Franke and S. Heinrich, "Fast obstacle detection for urban traffic situations," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 3, pp. 173–181, Sep. 2002.

[29] T. Dang, C. Hoffmann, and C. Stiller, "Fusing optical flow and stereo disparity for object tracking," in *Proc. IEEE 5th Int. Conf. Intell. Transp. Syst.*, 2002, pp. 112–117.

[30] X. Li, X. Yao, Y. Murphey, R. Karlsen, and G. Gerhart, "A real-time vehicle detection and tracking system in outdoor traffic scenes," in *Proc. 17th Int. Conf. Pattern Recog.*, 2004, pp. II:761–II:764.

[31] R. Danescu, S. Nedevschi, M. Meinecke, and T. Graf, "Stereovision based vehicle tracking in urban traffic environments," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2007, pp. 400–404.

[32] A. Zomotor, *Fahrwerktechnik/Fahrverhalten*, 1st ed. Stuttgart, Germany: Vogel, 1987.

[33] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6D-vision: Fusion of stereo and motion for robust environment perception," in *Proc. 27th DAGM Symp.*, 2005, pp. 216–223.

[34] S. Gehrig and F. Stein, "Collision avoidance for vehicle-following systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 233–244, Jun. 2007.

[35] A. Polychronopoulos, M. Tsogas, A. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 549–562, Sep. 2007.

[36] N. Kaempchen, K. Weiss, M. Schaefer, and K. Dietmayer, "IMM object tracking for high dynamic driving maneuvers," in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 825–830.

[37] C. Rabe, U. Franke, and S. Gehrig, "Fast detection of moving objects in complex scenarios," in *Proc. IEEE Intell. Veh. Symp.*, 2007, pp. 398–403.

[38] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, Apr. 1991.

[39] M. Bühren and B. Yang, "Initialization procedure for radar target tracking without object movement constraints," in *Proc. 7th Int. Conf. ITS Telecommun.*, 2007, pp. 112–117.

[40] C. McGlone, Ed., *Manual of Photogrammetry*, 5th ed. Bethesda, MD: Amer. Soc. Photogramm., 2004.

[41] H. Badino, T. Vaudrey, U. Franke, and R. Mester, "Stereo-based free space computation in complex traffic scenarios," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation*, 2008, pp. 189–192.

Alexander Barth was born in Lüdenscheid, Germany, in 1980. He received the B.Sc. and M.Sc. degrees in computer science from the University of Applied Sciences of Bonn, Bonn, Germany, in 2003 and 2005, respectively. He is currently working toward the Ph.D. degree in computer science with the University of Bonn, in cooperation with the Environment Perception Group, Daimler AG, Sindelfingen, Germany.

His research interests are in the areas of motion analysis, traffic scene understanding, object tracking, and stereo vision.

Uwe Franke received the Ph.D. degree in electrical engineering from the Technical University of Aachen, Aachen, Germany, in 1988 for his work on content-based image coding.

Since 1989, he has been with Daimler AG, Sindelfingen, Germany, where he has been working on vision-based driver-assistance systems. He developed lane keeping and Daimler's lane-departure warning system (Spurassistent) and has been working on sensor fusion and truck platooning. His special interest is in image understanding in complex situations like driving in cities and on real-time stereo analysis. His recent work has concerned optimal fusion of stereo and motion, which is called 6D-Vision and scene flow. Since 2000, he has been the Head of the Daimler Environment Perception Group and concentrates on vision for increased traffic safety. Continuously working in the field of intelligent vehicles for the past 19 years, he is one of the most experienced experts in the world.

Dr. Franke was the Program Chair of the 2002 IEEE Intelligent Vehicles Conference in Versailles, France.