Tracking Oncoming and Turning Vehicles at Intersections

Alexander Barth and Uwe Franke

***Abstract*— This article addresses the reliable tracking of oncoming traffic at urban intersections from a moving platform with a stereo vision system. Both motion and depth information is combined to estimate the pose and motion parameters of an oncoming vehicle, including the yaw rate, by means of Kalman filtering. Vehicle tracking at intersections is particularly challenging since vehicles can turn quickly. A single filter approach cannot cover the dynamic range of a vehicle sufficiently.**

We propose a real-time multi-filter approach for vehicle tracking at intersections. A gauge consistency criteria as well as a robust outlier detection method allow for dealing with sudden accelerations and self-occlusions during turn maneuvers. The system is evaluated both on synthetic and real-world data.

I. INTRODUCTION

In 2008, 15.7% of all road accidents in Germany with damage to persons happened at intersections and during turning maneuvers [1]. Intersections are accident hot spots and, thus, of special interest for future driver assistance systems. Detecting and tracking vehicles at intersections with stationary cameras, typically from elevated position, has been addressed by many researchers in the past, e.g., [2], [3], [4].

Previous work on vision-based vehicle tracking from a moving platform mainly concentrates on highway scenarios. However, precise information on the behavior of the oncoming and cross traffic at intersections provides a fundamental basis for future driver assistance and safety applications. The pose and motion state of oncoming vehicles, relative to the ego vehicle, is an essential information for high-level situation analysis, e.g., collision risk prediction for active collision avoidance systems.

In [5], we have proposed a generic system for vehicle tracking, in which objects are modeled as rigid 3D point clouds moving along circular paths. An extended Kalman filter is used for estimating the pose and motion parameters, including velocity, acceleration, and rotational velocity (yaw rate). In [6], we have extended this feature-based approach by geometrical constraints that require the estimated object pose to be consistent with object silhouettes derived from dense stereo data.

The system model of the Kalman filter allows for modeling a particular expected dynamic behavior of a tracked instance. At intersections, there are typically two options: Straight motion or turning. The former is best modeled by a stationary process, e.g., constant velocity linear motion, while turning vehicles require higher-order motion models that allow, for example, to quickly develop a yaw rate. A system model designed for mainly stationary processes may be too slow to follow a maneuvering (turning) target, on the other hand a dynamic model in general may be too reactive to errors in the observations.

A. Barth is with the University of Bonn, Germany, Institute of Geodesy and Geoinformation. alexander.barth@uni-bonn.de

U. Franke is with Daimler AG, Group Research & Advanced Engineering, Sindelfingen, Germany. uwe.franke@daimler.com

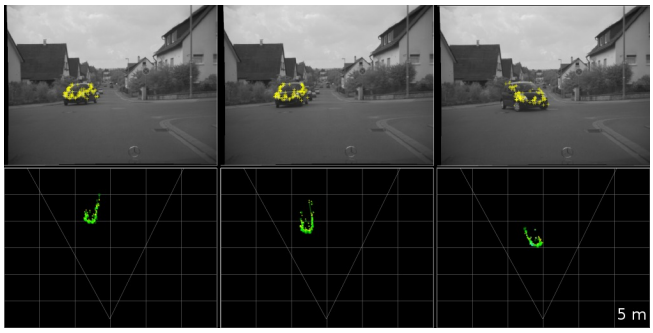


Fig. 1. At intersections, a vehicle tracking system must be able to deal with highly dynamic turn maneuvers and self occlusions. The tracked object points building the object model are superimposed in the images (top) and shown from bird's eye view (bottom).

Several solutions to deal with maneuvering targets have been proposed in the literature, including input estimation [7], adaptive system noise control [8], and multi-filter approaches [9]. We have compared these approaches on simulated data in [10]. The Interacting Multiple Model (IMM) approach of Bar-Shalom has resulted in the best compromise between computation time and tracking performance on the considered test bed. Kaempchen et al. [11] have successfully applied the IMM framework to track leading vehicles in stop-and-go scenarios.

Beside the choice of the right dynamic model, tracking of quickly turning vehicles requires dealing with changing visibility of object points. During the turn maneuver, some object points become occluded, as the orientation of the tracked vehicle changes, while others become visible and have to be registered to the object model. An example on the problem at hand is shown in Fig. 1.

In this contribution, we propose an IMM-based filtering approach for vehicle tracking at urban intersections. The system is able to automatically choose the right motion model for typical urban scenarios. We extend the measurement model to ensure a stable gauge definition by adding additional constraints on the point cloud. Furthermore, we introduce an adaptive outlier detection mechanism that is able to prevent that all points are detected as outliers when the vehicle enters the turn maneuver.

The tracking performance is evaluated based on real-world scenes as well as realistic virtual scenes (rendered images) with ground truth available.

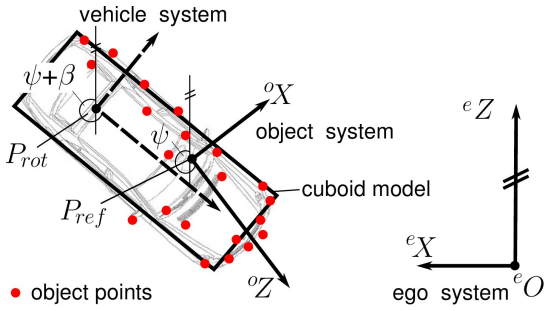


Fig. 2. Coordinate systems from bird's eye view.

This article will be organized as follows. In Section II we will summarize the generic vehicle tracking framework and propose several intersection specific extensions. The design of the multi-filter approach is given in Section III. Experimental results are presented and discussed in Section IV, followed by our conclusions in Section V.

II. VEHICLE TRACKING APPROACH

One filter cycle consists of three parts: First, the object state is predicted based on an *object shape and motion model* (Kalman filter prediction step). Then, the predicted state is updated based on a *measurement model*, incorporating depth and motion information from the stereo vision system (Kalman filter update step). Finally, the object shape model is refined outside the filter. In the following we will explain the different parts in more detail.

A. Object Model

In our approach, objects are modeled by a rigid 3D point cloud $\boldsymbol{\Theta}$, with $\boldsymbol{\Theta} = [{}^o\boldsymbol{P}_1, {}^o\boldsymbol{P}_2, \ldots, {}^o\boldsymbol{P}_M]^\mathsf{T}$ representing the object's shape, and a surrounding cuboid approximating the object dimensions with parameters $\boldsymbol{D} = [w, l, h]^\mathsf{T}$, corresponding to the width, length, and height, respectively. Fig. 2 gives an overview on the object model and the defined coordinate systems. Decoupling the object dimension from the point cloud is especially beneficial at intersections, since the observable object point cloud might be incomplete, e.g., due to visibility constraints or partial occlusions. It further enables opportunities for sensor fusion of extended objects, e.g., with lidar data.

1) Gauge Definition: We attach a local object coordinate system (at theoretically arbitrary position) to the object point cloud, which defines the pose of a rigid body. In our approach, the centroid and main principal axis of the initial point cloud are used. Each point of the rigid point cloud has a fixed position within the object coordinate system. This *gauge* (or *datum*) definition allows for inferring the object pose at consecutive time steps based on observations of the point cloud. One has to ensure that the gauge definition is consistent over time, to be able to deal with a varying number of objects points, e.g., due to changing visibility during turn maneuvers. This will be addressed in more detail in Section II-C.

The pose of the object coordinate system with respect to the ego vehicle is fully defined by six parameters (3D translation and 3D rotation). However, only the lateral and longitudinal coordinate of the object origin in *ego coordinates* (a coordinate system attached to the ego vehicle), denoted as reference point ${}^e\boldsymbol{P}_\text{ref} = [{}^eX_\text{ref}, {}^e0, {}^eZ_\text{ref}]^\mathsf{T}$, as well as the rotation around the height axis by angle ψ are estimated, as this is the only rotational parameter that can be controlled by the driver using the steering wheel.

2) Motion Model: Movements are restricted to circular path motion based on a simplified bicycle motion model, since road vehicles typically cannot move sidewards. This motion model is parametrized by velocity v and acceleration $\dot{v}$ in the moving direction as well as the yaw rate $\dot{\psi}$, i.e., the change of orientation. Straight motion can be thought of as driving on a circle with infinite radius. To be able to deal with sudden changes of the yaw rate, we also incorporate the yaw acceleration $\ddot{\psi}$ into our motion model, assuming constant yaw acceleration instead of constant yaw rate. In general we can write the motion model as nonlinear function f, with

$$\widehat{\boldsymbol{x}}^-(k) = f\left(\widehat{\boldsymbol{x}}^+(k-1), \boldsymbol{u}(k)\right) + \boldsymbol{\omega}, \tag{1}$$

i.e., the a priori state $\widehat{\boldsymbol{x}}^-$ estimate at discrete time k is derived from the a posteriori state estimate $\widehat{\boldsymbol{x}}^+$ at time $k-1$ and some control input vector $\boldsymbol{u}$ incorporating, for example, the ego motion. Higher-order terms are modeled as additive zero-mean white Gaussian noise process $\boldsymbol{\omega}$ with covariance matrix $\boldsymbol{C}_{\omega\omega}$ (*system noise matrix*).

The motion model further requires the object origin to be located at the center rear axle of the vehicle, which we will denote as *rotation point*, $\boldsymbol{P}_\text{rot} = [{}^oX_\text{rot}, 0, {}^oZ_\text{rot}]^\mathsf{T}$, in the following, and the moving direction $\chi = \psi + \beta$ to be aligned with the longitudinal axis (Z-axis) of the vehicle. Since both the relative position of the rotation point to the reference point (object origin) and the moving direction are typically not known at initialization, these parameters have to be estimated.

3) Filter Representation: The pose and motion parameters are summarized in the following state vector:

$$\boldsymbol{x} = \left[\underbrace{{}^eX_\text{ref}, {}^eZ_\text{ref}, \psi}_{\text{pose}}, \underbrace{{}^oX_\text{rot}, {}^oZ_\text{rot}, \beta, v, \dot{v}, \dot{\psi}, \ddot{\psi}}_{\text{motion}}\right]^\mathsf{T}. \tag{2}$$

With the parameters ${}^oX_\text{rot}$, ${}^oZ_\text{rot}$, and β it is possible to transform any point in object coordinates into a virtual vehicle coordinate system with the origin at the center rear axle and the longitudinal axis corresponding to the moving direction.

4) Initialization: Object points are detected and grouped based on the Gestalt principle of common fate, i.e., points of common motion are likely to belong to the same object. This idea is used both for initialization of new object tracks and for adding new points to existing object models. We independently track the 3D position and 3D velocity of a number of good-to-track image features, distributed over the whole image plane, based on the 6D-Vision principle [12].

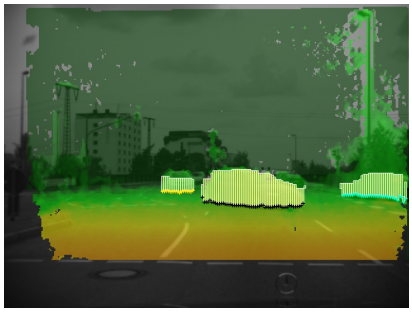


Fig. 3. The stixel world [14] is a compact 3D representation for traffic scenes, including information on free space and obstacles. The color coding here is red ↔ green as close ↔ far.

Clusters of 6D vectors with common motion generate object hypotheses. The average velocity vector gives the initial moving direction and the initial vehicle speed. Both the reference point and the rotation point are initialized at the centroid of the initial point cloud. At the same time, tracked image features, whose 3D position fall into the cuboid model of an existing object, are added to the object's shape model, if also compatible in motion.

B. Measurement Model

The measurement model consists of two parts: Direct measurements of the (sparse) object points and geometric measurements of the rotation point, derived from dense stereo vision.

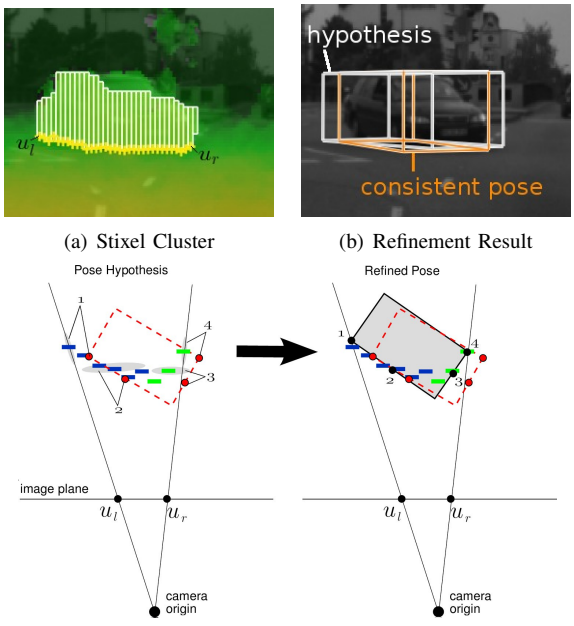
1) Point Measurements: Each object point ${}^o\boldsymbol{P_m}$ with ${}^o\boldsymbol{P_m} = [{}^oX_m, {}^oY_m, {}^oZ_m]^\mathsf{T} \in \boldsymbol{\Theta}$, $1 \leq m \leq M$, is observed in terms of an image coordinate $\langle u_m(k), v_m(k) \rangle$ and stereo disparity $d_m(k)$ at time k. These measurements are summarized in $\boldsymbol{z}_m(k) = [u_m(k), v_m(k), d_m(k)]^\mathsf{T}$.

The projection of each point onto the image plane is tracked using a feature tracker, e.g., the well-known KLT-tracker [13], to be able to reassign measurements of the same 3D point over a sequence of images. The nonlinear measurement model h, from which the predicted measurements $\widehat{\boldsymbol{z}}_m = h(\widehat{\boldsymbol{x}}^-, {}^o\boldsymbol{P_m})$ are derived, directly follows from the transformation between object and camera coordinates and the well-known projection equations of a finite perspective camera (see [5] for details).

2) Rotation Point Measurements: A good estimate of the rotational center, located at the center rear axle of the vehicle, is essential for the prediction of the point positions from one to the next discrete time step. The rotational center, however, is only weakly observable from the point movements during turn maneuvers and not at all while the vehicle is moving straight. The idea is to stabilize the rotation point estimate by measuring its position geometrically.

The cuboid model is used to integrate a basic semantical meaning on vehicle sides (front, rear, left, right) or characteristic points, e.g., front left corner or center of the right side, into the, beside rigidity, unconstrained point cloud model.

Given an object hypothesis, i.e., cuboid pose and dimension D, the objective is to refine the parameters of the cuboid to yield an estimate which is consistent with the information



(a) Stixel Cluster (b) Refinement Result

(c) Geometric Constraints

Fig. 4. Several constraints on the object pose and boundaries, including the viewing angle and distance of the most outer bounding box corners and the centers of the visible sides, are derived from a stixel cluster. The initial pose and cuboid dimension is iteratively refined by a maximum likelihood estimation. See [6] for details.

available from dense stereo depth maps. We use a real-time FPGA implementation of the semi-global matching (SMG) algorithm as proposed in [15].

For real-time applicability we do not directly work on the per pixel depth information, but utilize an efficient intermediate representation, denoted as *Stixel World* [14], that models the 3D environment in terms of so called *stixel* entities. Fig. 3 shows a stixel world example for a typical intersection scene. Each stixel contains information on distance, height, and viewing angle with respect to the camera center. This compact representation covers both information on the drivable free space and obstacles. The main principle of the pose refinement is visualized in Fig. 4.

From the refined object pose we derive a pseudo-measurement of the rotation point position, defined at the lateral center of the cuboid at a constant distance to the rear side. It is integrated into the Kalman filter as direct measurement of the rotation point.

3) Outlier Detection: As all mean-squares like estimation techniques the Kalman filter is quite sensitive to outliers in the measurements. Thus, outliers have to be detected and removed from the measurement vector before an update. If the measurements are uncorrelated, an outlier test can be performed for each measurement separately based on the normalized innovation squared (NIS) measure:

$$\delta_m = \boldsymbol{r}_m^\mathsf{T} \left(\boldsymbol{C}_{rr}^{(m)} \right)^{-1} \boldsymbol{r}_m \tag{3}$$

where $\boldsymbol{r}_m = \boldsymbol{z}_m - \widehat{\boldsymbol{z}}_m$ indicates the residual between actual and predicted measurement of point ${}^o\boldsymbol{P}_m$ and $\boldsymbol{C}_{rr}^{(m)}$ the

corresponding 3×3 covariance matrix. The square root of this measure corresponds to the Mahalanobis distance and expresses the residual in terms of standard deviations.

If $\sqrt{\delta_m} > \sigma_{\max}$, e.g., $\sigma_{\max} = 3$, this measurement is rejected as outlier. A robust, M-estimator like reweighing of points that pass this test is applied to reduce the influence of remaining outliers in the data as proposed in [16].

Outlier detection during a maneuvering phase requires a special consideration. At the beginning of the maneuver, larger deviations between prediction and measurements occur. The filter has to compensate for that deviation by altering the object pose and motion parameters. However, as these deviations can be significant if a vehicle starts to turn quickly, all points might be detected as outliers and the tracking cannot be continued.

To prevent this problem, we introduce an adaptive outlier threshold. Instead of considering each point independently, the group of points is evaluated. Assuming there are not more than 50% outliers in the data, we replace the constant threshold $\sigma_{\max}$ above by $\sigma'_{\max}$ with

$$\sigma'_{\max} = \max\left(\sigma_{\max}, \text{median}(\sqrt{\delta_1}, \ldots, \sqrt{\delta_M}\right). \quad (4)$$

This formulation guarantees that at least half of all points survive the outlier test.

C. Shape Model Update

Since the exact position of a given object point in the shape model is typically not known at initialization, it has to be estimated from the noisy measurements. For real-time applicability, the problem of motion estimation is separated from the problem of shape reconstruction. Thus, instead of estimating shape and motion simultaneously by integrating $\boldsymbol{\Theta}$ into $\boldsymbol{x}$, the point positions are refined outside the Kalman filter in two steps.

First, each point is updated independently of all other points based on its measured position, yielding an estimate ${}^o\widehat{\boldsymbol{P}}^*_m(k)$. Then, all updated point positions are corrected by a common rigid transformation to the posterior estimated position ${}^o\widehat{\boldsymbol{P}}_m(k)$, ensuring that the centroid and the principal axis of the point cloud are not changed by the individual updates as will be motivated in more detail below.

Step 1: Assuming uncorrelated measurements over time, a maximum likelihood estimate for ${}^o\boldsymbol{P}_m(k)$ is given by

$${}^o\widehat{\boldsymbol{P}}^*_m(k) = \left[\sum_{\kappa=k_m}^{k} C_m^{-1}(\kappa)\right]^{-1} \sum_{\kappa=k_m}^{k} C_m^{-1}(\kappa)\, {}^o\tilde{\boldsymbol{P}}_m(\kappa) \quad (5)$$

where $C_m(k)$ denotes the 3×3 covariance matrix of the measured point ${}^o\tilde{\boldsymbol{P}}_m(k)$, and k_m the discrete time step the m-th point has been added to the model.

Step 2: Let Q denote an index set of points that have been observed at two consecutive time steps and not detected as outlier, with $Q \subseteq \{1 \ldots M\}$. Then, each updated point ${}^o\widehat{\boldsymbol{P}}^*_q(k)$ is corrected as follows:

$$\begin{aligned} {}^o\widehat{\boldsymbol{P}}_q(k) =& \boldsymbol{R}_y^{-1}(\Delta_\theta)\left({}^o\widehat{\boldsymbol{P}}^*_q(k) - {}^o\bar{\boldsymbol{P}}(k)\right) \\ &+ {}^o\bar{\boldsymbol{P}}(k-1), \quad \forall q \in Q. \end{aligned} \quad (6)$$

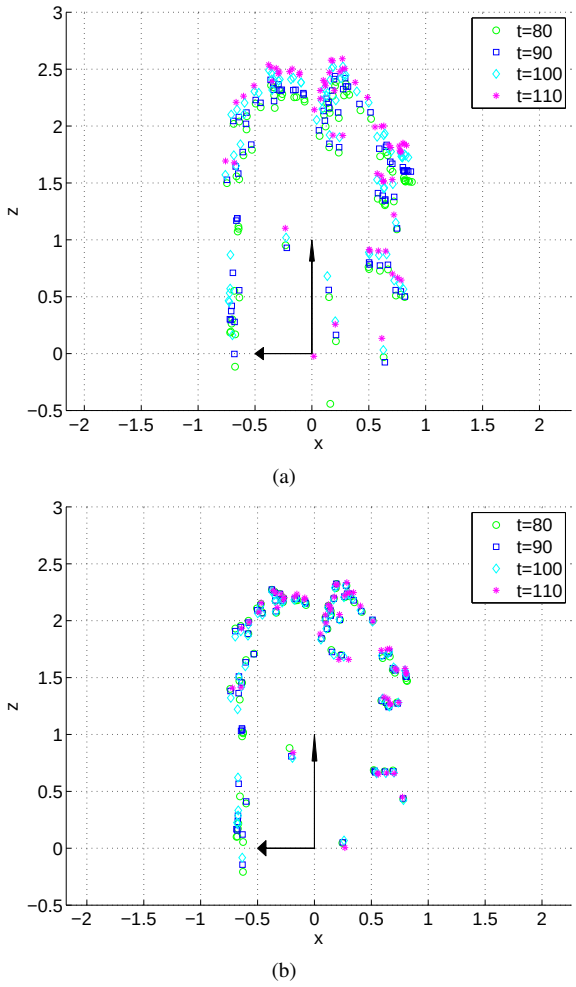


Fig. 5. Object points from bird's eye view at four time steps. (a) If all object points are updated independently by weighted averaging, they drift in the moving direction as the vehicle accelerates. (b) The gauge consistency correction successfully prevents a point drift.

with ${}^o\bar{\boldsymbol{P}}(k) = \frac{1}{|Q|}\sum_{q\in Q} {}^o\widehat{\boldsymbol{P}}^*_q(k)$ the mean of the points in Q at time step k and $\boldsymbol{R}_y(\Delta_\theta) \in \mathbb{R}^{3\times3}$ a rotation about the height axis by angle Δ_θ. This angle is defined as the difference between the orientation of the main principal component of the points in Q at time k and $k-1$. At this step it is assumed that the rotation between both point clouds does not change more than $\pi/2$ rad. This global correction ensures that the sum of all point updates is zero (*centroid consistency*) and that the orientation of the principal component of the adapted point cloud is equal to the orientation of the previous point cloud in object coordinates (*principal component consistency*). It prevents that points can systematically drift within the object coordinate system in situations where the predicted object pose differs considerably from the actual pose, without changing the pose and motion parameters.

The effect is demonstrated in Fig. 5. In this example, the tracked vehicle is accelerating. As can be seen in (a), updating the point positions within the local object coordinate system by weighted averaging, leads to a drift in the movement

direction (z-axis). This indicates that the estimated velocity is too low. The gauge correction in (6) prevents the drift, i.e., systematic errors between the predicted and measured point positions must be mainly compensated by altering the motion parameters in the filter, instead of changing a point's relative position with respect to the local object coordinate system.

The cuboid dimension is also updated outside the Kalman filter by lowpass filtering of the estimated dimensions from the stixel silhouette analysis.

III. MULTI-FILTER DESIGN

A. Interacting Multiple Models (IMM)

The IMM framework fuses a set of r filters, where each filter represents a certain *mode*, e.g., non-maneuvering or maneuvering. One IMM filter cycle consists of three parts: First, the r *a posteriori* state estimates and covariance matrices of the previous discrete time step are probabilistically *mixed* by weighted averaging (interaction step). Then, the filters are updated based on a common measurement vector (filtering step). The mixing prevents a combinatorial explosion, since there are r^k possible filter configurations at time k if all r filters are run with all filter outputs from previous time steps. Finally, a mode probability is computed for each filter based on the normalized residual between prediction and measurements (mode probability update step). Details on the IMM framework can be found in [17].

The linear mixing of states and covariance matrices, as proposed in the original approach, cannot be applied to our model, since the state vector contains Euler angles. Thus, we use an alternative nonlinear formulation that handles ψ and β separately.

B. Motion Model Selection

At intersections, vehicles show a variety of different movement patterns that a tracking system must be able to deal with, e.g., starting, stopping, passing, or turning. During these movements there are *stationary* (constant) phases and *maneuvering* phases, indicated by significant changes in the dynamics.

The driver has mainly two options to influence the trajectory: Altering the velocity via the gas pedal (or brake) and controlling the yaw rate by turning the steering wheel. To cover all possible driving states by a small number of discrete motion models, we summarize all continuous configurations of the gas pedal and steering wheel into the discrete states *constant* and (constantly) *accelerated*. The permutations of these states yield four classes of movements:

- constant velocity / constant yaw rate (CVCY)
- accelerated velocity / constant yaw rate (AVCY)
- constant velocity / accelerated yaw rate (CVAY)
- accelerated velocity / accelerated yaw rate (AVAY)

The first two models (and further simplifications such as constant velocity / constant yaw angle) are commonly used in the literature for vehicle tracking [18]. The yaw acceleration is typically modeled as a Gaussian noise process. At turn maneuvers, however, the yaw acceleration becomes a significant issue. In [10], we have shown that incorporating the yaw acceleration into the state vector of a single filter approach significantly improves the tracking performance at turn maneuvers. However, the system becomes more sensitive to non-detected outliers in the data, leading to instabilities at straight-line motion.

	σ_{pos}	$\sigma_\psi, \sigma_\beta$	$\sigma_{\dot\psi}$	σ_v	$\sigma_{\dot v}$	$\sigma_{\ddot\psi}$
$\boldsymbol{C}_{\omega\omega}^{(\text{stat})}$	0.01	0.01	0.01	0.01	0.1	-
$\boldsymbol{C}_{\omega\omega}^{(\text{mnv})}$	0.01	0.01	0.1	0.1	2	0.5

TABLE I
FILTER PARAMETER ($\boldsymbol{C}_{\omega\omega} = diag(\sigma_{\text{POS}}^2, \sigma_{pos}^2, \sigma_{\dot\psi}^2, \sigma_{\dot v}^2, \dots)$)

In this approach, we overcome this problem by combining the AVAY model with an AVCY model via the IMM framework to a two filter approach. The AVCY model ignores the estimated yaw acceleration and, thus, is insensitive to errors in this parameter. The CVCY and the CVAY model are not considered in our approach to reduce the complexity of the filter with respect to real-time applicability. Most trajectories involve at least some amount of acceleration/deceleration, i.e., constant velocity models are unlikely at intersections.

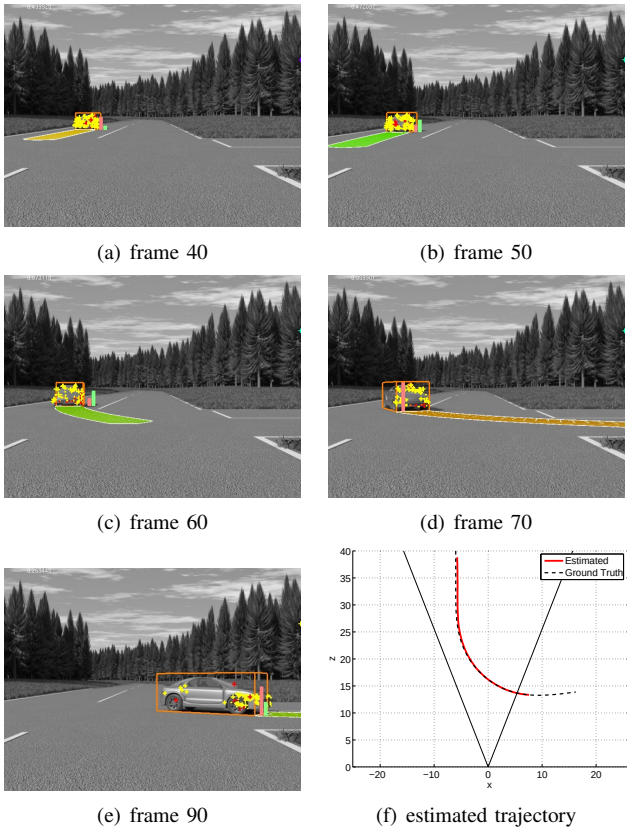
Combining filters with different motion models is superior to approaches where each filter uses the same motion model and the filter behavior is controlled via the system noise matrices (one filter with small variances, one with large variances) [17]. The advantage is that different models are able to generate more distinctive, competing hypotheses at the prediction step.

C. Filter Configuration

The AVCY model (mode 1) is parametrized in a way that it covers mainly stationary processes, while the AVAY model (mode 2) is designed for more dynamic maneuvers, especially turn maneuvers. The system noise matrix of the AVCY model, $\boldsymbol{C}_{\omega\omega}^{(\text{stat})}$, allows for slow changes in accelerations ($\sigma_{\dot v} = 0.1$ m/s^2) and minor changes of the yaw rate ($\sigma_{\dot\psi} = 0.01$ rad/s).

Upon starting, vehicles accelerate with approximately 1.5 to 3 m/s^2, and stop with -1.5 to -5 m/s^2. We thus allow changes of the acceleration by 2 m/s^2 via the AVAY system noise matrix $\boldsymbol{C}_{\omega\omega}^{(\text{mnv})}$. This is a compromise between the maximum expected acceleration and the objective to yield smooth velocity estimates (constant acceleration). The allowed changes of the yaw rate are increased to 0.1 rad/s in the dynamic model. In addition, the yaw rate depends on the estimated yaw acceleration, which is able to change by 0.5 rad/s^2 in our configuration. The exact filter parameterization is given in Table I (here σ_{pos} covers both the reference and the rotation point entries of the state vector).

The IMM framework further requires the specification of mode transition probabilities, i.e., how likely does the filter switch from one mode to another. We use the following state transition matrix:



(a) frame 40 (b) frame 50 (c) frame 60 (d) frame 70 (e) frame 90 (f) estimated trajectory

Fig. 6. IMM tracking results of synthetic intersection scene. The estimated motion state allows for accurate prediction of the object's driving path. The estimated trajectory (red solid line) precisely fits the ground truth (dashed black line) as shown from bird's eye view on the bottom right.

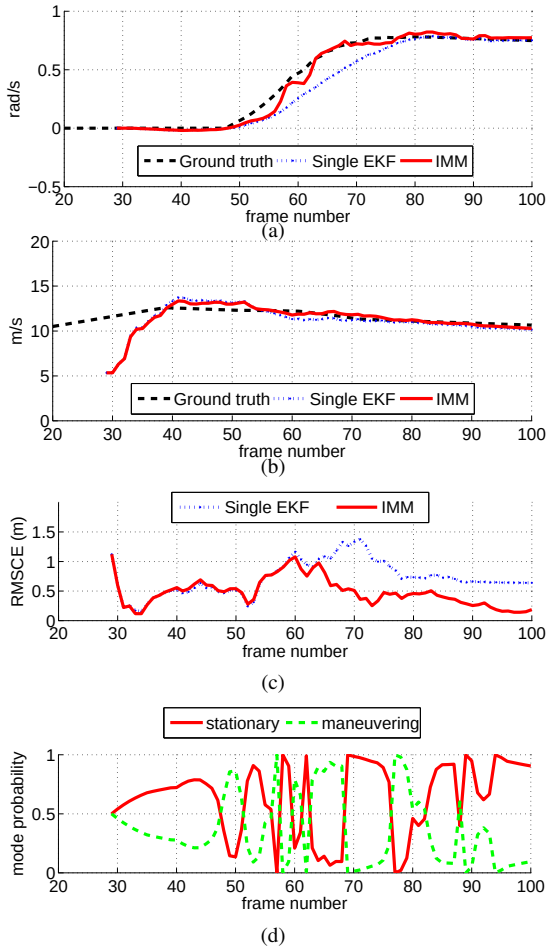


Fig. 7. Estimation results of IMM filter for (a) yaw rate and (b) velocity compared to ground truth (dashed black lines) and a single filter approach (dotted blue line). The pose accuracy is evaluated based on the root mean squared corner error in (c). The IMM approach clearly outperforms the single filter approach. The mode probabilities are shown in (d).

$$\Pi = \begin{bmatrix} 0.98 & 0.02 \\ 0.10 & 0.90 \end{bmatrix} \tag{7}$$

The entry at the i th row and j th column indicates the transition probability from mode i to j. For better robustness, this configuration slightly prefers the stationary mode, since it is less sensitive to the (noisy) measurements.

IV. EXPERIMENTAL RESULTS

In the following experiments, a stereo vision system with 0.3 m baseline and approximately 40 degrees viewing angle is used. The capture rate is 25 fps (VGA images).

A. Simulation Results

The proposed system is tested on a realistic synthetic stereo image sequence with ground truth information on the pose and motion parameters available. The scenario contains an oncoming vehicle approaching from 60 m distance and quickly turning to the left at approximately 15 m distance in front of the stationary ego vehicle. The velocity ($\approx$ 10 m/s) is slightly decreased during the turn maneuver.

Fig. 6 shows selected frames of the sequence with the estimation results superimposed. The object pose and dimension is indicated by the bounding box, the motion parameters are encoded in the predicted driving path for the next second, visualized as a carpet on the ground. The tracked feature points, building the object shape model, are visualized by yellow crosses (outliers are marked by red crosses).

The object is detected at about 40 m distance in frame 29. The maneuver starts at frame 50. The predicted driving path already starts to bend at this time step, although the vehicle has not significantly changed its orientation. Ten frames (=400 ms) ahead, the predicted driving path clearly indicates the vehicle is turning to the left. Its destination is accurately predicted at frame 70 as shown in frame 90.

As can be seen in Fig. 7(a) and (b), the estimated motion parameters approximate the ground truth very well. The estimation result of a single filter approach with AVCY motion model and manually optimized filter parameters is also shown for comparison. During the straight motion phase the single filter performs equally well as the IMM filter, however, it cannot follow the sudden increase of the yaw rate as the maneuver starts. The root-mean-squared-error (RMSE) of the yaw rate is 0.0443 for the IMM and 0.1073 for the single filter approach. The RMSE in velocity is 1.0724 (0.3985) for the IMM and 1.1273 (0.5802) for the single EKF respectively. The values in brackets indicate the RMSE

if the first 10 frames are ignored. This reduces the influence of the significantly underestimated velocity at initialization.

The tracking accuracy is evaluated based on the distance of the four corners of the estimated object bounding box compared to the ground truth corner positions in ego coordinates. The corners contain information on the object pose and dimension. Fig. 7(c) shows the root mean squared corner error (RMSCE), which decreases over time to below 0.15 m (average 0.49 m) for the IMM. The single filter approach results in a significantly larger error during the whole turn maneuver (average 0.72 m).

The IMM mode probabilities in Fig. 7(d) indicate the *interaction* of the two modi. Instead of switching once from stationary to maneuvering mode at the beginning of the turn maneuver, the system toggles between the two motion models to yield a combined motion model that allows for sudden changes in the motion parameters without loosing smoothness. A good example for this is the yaw rate estimate. Climbing the ramp in a few *stairs*, induced by short phases of yaw acceleration followed by a constant yaw rate, is much faster compared to the single filter approach assuming a constant yaw rate for all frames.

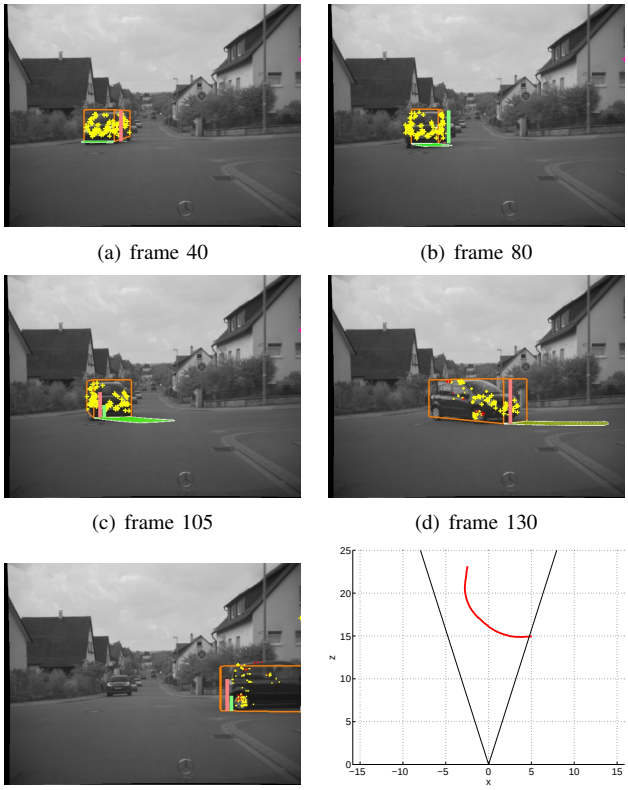
The proposed IMM filter switches to maneuvering mode only shortly to adapt the motion parameters, and then quickly falls back to the less reactive stationary mode. This property prevents the system becoming too reactive to the noisy measurements for a longer period, i.e., it increases the robustness of the approach and ensures smooth trajectory estimates.

B. Real-World Results

The tracking results of the intersection scene in Fig. 1 are shown in Fig. 8. It contains an oncoming vehicle stopping at the intersection before it turns to the left. The scene is particularly challenging, since the vehicle turns through a small radius and there is both an acceleration in the longitudinal direction as well as a sudden increase of the yaw rate. As in the simulation, there are also self occlusions during the turn maneuver.

As can be seen, the system is able to accurately track the object during the whole maneuver. The estimated pose parameters are shown by the bounding boxes, the estimated motion parameters are shown in Fig. 9.

At the beginning, the filter is in stationary mode (constant negative acceleration) and toggles to maneuvering mode only shortly at frame 60 to reduce the acceleration as the vehicle stops. At the onset of the turn maneuver, the filter switches again to maneuvering mode at about frame 75 to be able to quickly develop an acceleration in the longitudinal direction as well as with the yaw rate. At about frame 105, the constant yaw rate model outperforms the constant yaw acceleration model, leading to another mode change. This is typically the point where the driver shifts up into the next gear, interrupting the acceleration for a short period (about one second). The stationary filter only allows for slow changes in acceleration, thus, the velocity is slightly underestimated for a couple of frames. At frame 160, the



(a) frame 40 (b) frame 80 (c) frame 105 (d) frame 130 (e) frame 160 (f) estimated trajectory

Fig. 8. Tracking results of a vehicle turning through a small radius after stopping.

probability for maneuvering mode increases again, on the one hand to increase the velocity by increasing the acceleration, on the other hand to reduce the yaw rate to zero as the vehicle is going back to straight motion while it leaves the viewing field of the camera.

Further tracking results, including straight-line movements, are shown in Fig. 10.

The processing time on a recent Intel Quad Core processor is 80 ms per frame (640×480 images), including 40 ms for the stixel computation, 25 ms for feature tracking and ego-motion computation, as well as $4 - 8$ ms for tracking a single object with a two filter configuration.

V. CONCLUSION

We have presented a real-time system for vehicle tracking at urban intersections. The results have shown the system is able to accurately estimate the pose and motion state of oncoming vehicles, including the yaw rate, yielding valuable information for trajectory prediction and future collision avoidance systems.

Two filters with different motion models are combined via the IMM framework, allowing for adequate modeling of movements with low and high dynamics. The system switches between the two models to automatically adapt the dynamics of the filter to the dynamics in the scene with no manual parameter tuning required. During a maneuvering phase both motion models *interact*, i.e., there are several

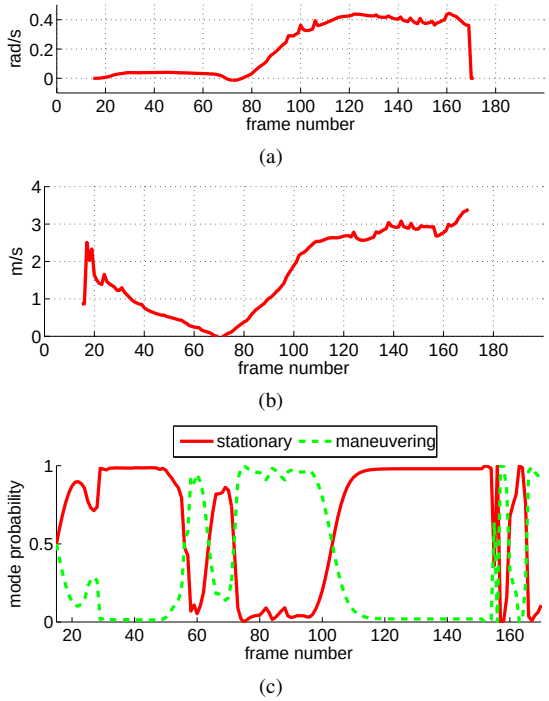


Fig. 9. Estimated motion parameters and IMM mode probabilities of the scene shown in Fig. 8. (a) yaw rate, (b) velocity, (c) IMM mode probabilities.

changes between the stationary and maneuvering mode. On the other hand, at straight-line motion the system remains in stationary mode for a longer time period.

In combination with the yaw rate and velocity estimate, this information provides useful information for turn maneuver detection. Classification of different movement patterns into straight-line motion and turn maneuvers is part of future work.

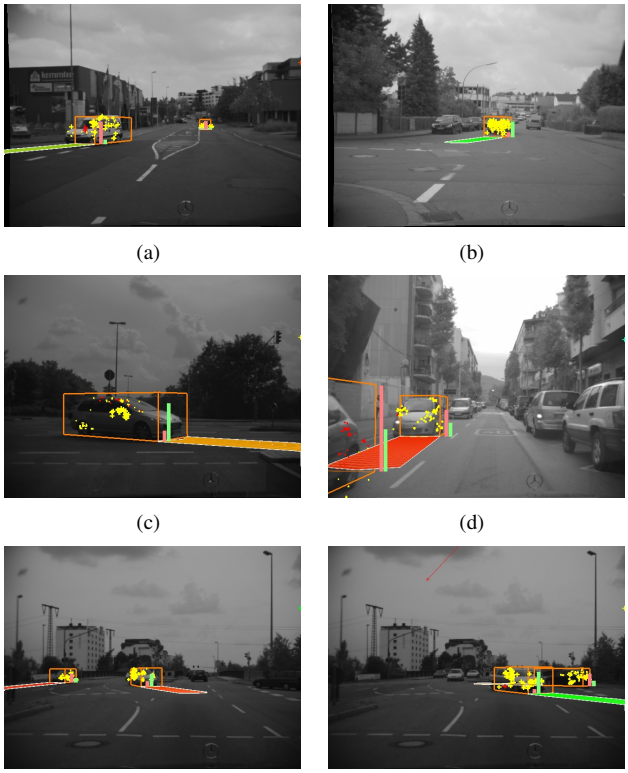


Fig. 10. Further tracking results of oncoming vehicles including turn maneuvers and straight-line motion. No manual adaptation of the filter parameters is required with the proposed system for all scenes in this figure.

REFERENCES

[1] Statistisches Bundesamt Wiesbaden, "Unfallgeschehen im Straßenverkehr 2008," 7 2009, German road traffic accident statistics.

[2] H. Veeraraghavan and N. Papanikolopoulos, "Combining multiple tracking modalities for vehicle tracking at traffic intersections," in *IEEE Conf. on Robotics and Automation*, 2004.

[3] S. Atev, H. Arumugam, O. Masoud, R. Janardan, and N. Papanikolopoulos, "A vision-based approach to collision prediction at traffic intersections," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 4, pp. 416–423, 12 2005.

[4] A. Ottlik and H. H. Nagel, "Initialization of model-based vehicle tracking in video sequences of inner-city intersections," *Int. J. Comput. Vision*, vol. 80, no. 2, pp. 211–225, 2008.

[5] A. Barth and U. Franke, "Estimating the driving state of oncoming vehicles from a moving platform using stereo vision," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 4, pp. 560–571, 2009.

[6] A. Barth, D. Pfeiffer, and U. Franke, "Vehicle tracking at urban intersections using dense stereo," in *3rd Workshop on Behaviour Monitoring and Interpretation, BMI*, Ghent, Belgium, 11 2009, pp. 47–58.

[7] Y. Chan, A. Hu, and J. Plant, "A kalman filter based tracking scheme with input estimation," *AES, IEEE Trans. on*, vol. 15, no. 2, pp. 237–244, 3 1979.

[8] S. J. Maybank, A. D. Worrall, and G. D. Sullivan, "A filter for visual tracking based on a stochastic model for driver behaviour," in *Proc. 4th European Conf. Computer Vision*. Springer-Verlag, 1996, pp. 540–549.

[9] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *Automatic Control, IEEE Transactions on*, vol. 33, no. 8, pp. 780–783, 8 1988.

[10] A. Barth, J. Siegemund, U. Franke, and W. Förstner, "Simultaneous estimation of pose and motion at highly dynamic turn maneuvers," in *DAGM Symposium on Pattern Recognition*, 2009, pp. 262–271.

[11] N. Kaempchen, K. Weiss, M. Schaefer, and K. Dietmayer, "IMM object tracking for high dynamic driving maneuvers," in *Intelligent Vehicles Symposium, IEEE*, 2004, pp. 825–830.

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

[13] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon University, Tech. Rep. CMU-CS-91-132, 4 1991.

[14] H. Badino, U. Franke, and D. Pfeiffer, "The stixel world - a compact medium level representation of the 3D world," in *DAGM Symposium on Pattern Recognition*, 9 2009.

[15] F. E. S. Gehrig and T. Meyer, "A real-time low-power stereo engine using semi-global matching," in *International Conference on Computer Vision Systems, ICVS*, 2009.

[16] Z. Zhang, Z. Zhang, P. Robotique, and P. Robotvis, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, pp. 59–76, 1997.

[17] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation with Applications To Tracking and Navigation*. John Wiley & Sons, Inc, 2001.

[18] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Information Fusion, International Conference on*, 2008, pp. 1–6.