Joint Vehicle Pose and Extent Estimation in the Context of Multi-Camera Traffic Surveillance

Leah Strand
Chair of Robotics, Artificial Intelligence and Real-time Systems
Technical University of Munich
Garching, Germany
leah.strand@tum.de

Jens Honer
Software Products and Services Product Line (DSW)
Valeo Schalter und Sensoren GmbH
Bietigheim-Bissingen, Germany
jens.honer@valeo.com

Alois Knoll
Chair of Robotics, Artificial Intelligence and Real-time Systems
Technical University of Munich
Garching, Germany
knoll@in.tum.de

***Abstract*—In this paper, we introduce a novel method for the estimation of vehicle pose and extent in traffic surveillance scenarios based on camera data. The state estimation is performed in a common world frame, enabling the seamless integration of the image data from different viewpoints. Our approach incorporates the non-linear transformation between the measurements and the states directly into the framework of an Unscented Kalman filter. Two measurement models are proposed: one designed for bounding boxes and another for discretized object contours extracted from segmentation masks. The method is evaluated using data from a real-world traffic surveillance system, demonstrating the high effectiveness and good feasibility of our approach for localizing passing cars.**

***Index Terms*—Vehicle Pose Estimation, Extent Estimation, Unscented Kalman Filter, Traffic Surveillance, Tracking.**

I. Introduction

Traffic surveillance systems can offer real-time situational awareness from an advantageous bird's eye view to drivers and autonomous vehicles through the creation of a digital twin of the road scene. Cameras, due to their high information content and simultaneous cost-effectiveness, have retained their popularity for implementing such systems. Nevertheless, the inherent difficulty in accurately estimating the real-world object state from a two-dimensional representation of the scene poses a significant challenge. While end-to-end learning-based pose estimation methods [1]–[4] show remarkable performance and are continually improving, they still lag in the ability to generalize to new camera positions and viewing angles. Notably, getting a model to generalize over diverse viewpoints requires a large amount of labeled 3D data. On the other hand, the ability of deep neural networks (DNNs) to effectively process image data is unmatched. Thus, hybrid approaches leveraging the capability of DNNs in combination with model-driven 3D reconstruction are widely investigated [5]–[11]. There, prior knowledge of the scene in the form of the camera calibration can be used to obtain the transformation between the respective camera and the world frame. In general, pose estimation is performed by matching the DNN observation to a virtual 3D model. The analyzed image observations are typically in the form of keypoints [5]–[8] or segmentation masks [9]–[11]. Most shape aware pose estimation approaches recover the 3D state of the objects by solving an optimization problem [5]–[9]. However, optimization-based methods are in general less favored for real-time and dynamic traffic surveillance applications with sequential data. In turn, Bayesian state estimation provides a suitable framework to estimate, predict and track the state of dynamic traffic participants over time. New sensor data is sequentially incorporated to refine the estimates at each time step while factoring in system uncertainties. In the work of Scheel *et. al* [10], a vehicle tracking approach is presented which fuses radar with semantic segmentation data within a random finite set-based multi-object framework. The measurement model is defined for the binary segmented image regarding each pixel as an independent measurement. The MaskUKF algorithm [11] employs an Unscented Kalman filter (UKF) for object pose tracking based on RGB-D data in the context of robotics manipulation tasks. There, a binary mask is segmented by a DNN to filter the relevant point cloud which is in turn matched to a 3D mesh model of an object to update its state.

The objective of our work is to fuse the data from a multi-camera traffic surveillance system with the task of localizing passing cars. Our approach performs joint pose and extent filtering in a common world frame by processing bounding boxes and segmentation masks as the observation input. We present a novel measurement model that defines the relationship between the object state and the measurement through the occupied area in the image space. Particularly, the binary segmented image is condensed into a discretized representation of the object contour. In this work, we focus on the measurement model itself and perform our evaluations based on single object tracking, however, with the prospective goal of establishing a system that provides a comprehensive representation of the traffic situation with all participants.

The paper is organized as follows: In Section II, we start with a concise introduction to Bayesian state estimation in general and the non-linear UKF in particular. Afterwards, we present our methodology in Sections III to V. We evaluate the methods in Section VI using data from a real-world system established during the research project *Providentia++* [12]–[14]. Finally, we summarize our results in Section VII.

This work was funded by the German Federal Ministry of Transport and Digital Infrastructure as part of the research project VIDETEC-2.

II. Preliminaries

In this section, we introduce the general concept of state estimation and in particular non-linear state estimation using the Unscented Kalman filter before describing the details of our method in the subsequent sections.

A. Bayesian State Estimation

Given a system with an unobservable internal state, state estimation provides an approach for inferring the system's actual state $\mathbf{x}_t \in \mathbb{X} \subseteq \mathbb{R}^n$ based on measurements $\mathbf{y}_t \in \mathbb{Y} \subseteq \mathbb{R}^N$ gathered at successive discrete points at time t. The object state is assumed to evolve according to a process function and is related to the measurement $\mathbf{y}_t$ by the measurement function $h\colon \mathbb{X} \mapsto \mathbb{Y}$. Both, the process transition and the measurement are assumed to be noisy or imperfect. Modeling the system state with a probability density function (pdf) to represent the element of uncertainty in the whole procedure, allows the problem to be formulated in terms of probabilistic concepts, in particular Bayesian inference. Using Bayes' theorem, the posterior distribution can be derived recursively from the prior pdf of the state as the conditional pdf given the obtained measurement. The relation is determined by the measurement likelihood $g(\mathbf{y}_t|\mathbf{x}_t)$ of observing a particular measurement $\mathbf{y}_t$ given the current state estimate $\mathbf{x}_t$ and the expected measurement noise. Assuming Gaussian distributions for both the prior and the likelihood, results in the posterior distribution to be Gaussian as well. This property entails a closed-form solution in the form of a set of algebraic equations for the mean and covariance of the normal distributions, respectively. These properties that only hold in the case of linear transition and measurement functions are also known as the Kalman filter. Yet, it is possible to approximate the relation for non-linear systems using for instance related filters like the Extended Kalman filter (EKF) or the Unscented Kalman filter (UKF).

B. Non-Linear State Estimation with the UKF

In this work, the measurement as well as the motion model are non-linear. Therefore, we employ the UKF which provides a derivative-free approach for non-linear state estimation by approximating the true state distribution through a set of deterministic sample points, known as sigma points [15], [16]. Since the steps of the UKF algorithm are carried out analogously for the process transition and the measurement, we outline only the measurement update in the following.

The $2n + 1$ sigma points $\mathcal{X}_i \in \mathbb{X}$ for a predicted state $\mathbf{x}_t^- \in \mathbb{X}$ with predicted covariance $P_t^- \in \mathbb{R}^{n\times n}$ are calculated following the formulas presented in [15], [16]. They are propagated through a non-linear function, in this case the measurement function, to yield the transformed points $\mathcal{Y}_i \in \mathbb{Y}$:

$$\mathcal{Y}_i = h(\mathcal{X}_i) \;\text{ for } 0 \leq i \leq 2n.$$

Here, the time index is omitted for brevity. Using suitable weights $W_i^m \in \mathbb{R}$, $W_i^c \in \mathbb{R}$ for $i = 0, \ldots, 2n$ and the measurement noise covariance $R \in \mathbb{R}^{N\times N}$, the mean $\bar{\mathbf{y}}$, the covariance P_{yy} and the cross-covariance P_{xy} of the transformed sigma points are computed as:

$$\begin{aligned}
\bar{\mathbf{y}} &= \sum_{i=0}^{2n} W_i^m \mathcal{Y}_i, \\
P_{yy} &= \sum_{i=0}^{2n} W_i^c (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T + R, \\
P_{xy} &= \sum_{i=0}^{2n} W_i^c (\mathcal{X}_i - \mathbf{x}^-)(\mathcal{Y}_i - \bar{\mathbf{y}})^T.
\end{aligned}$$

The method for choosing the weights published by Van der Merwe *et al.* [17] is widely established. Finally, the posterior mean and covariance are computed using the Kalman gain $K = P_{xy}P_{yy}^{-1}$ as a combination of the predicted state $\mathbf{x}_t^-$, the covariance P_t^-, and the measurement $\mathbf{y}_t$ as

$$\begin{aligned}
\mathbf{x}_t &= \mathbf{x}_t^- + K_t(\mathbf{y}_t - \bar{\mathbf{y}}_t), \\
P_t &= P_t^- - K_t P_{xy,t}^T.
\end{aligned}$$

C. Definition of an Object Contour

In addition, we introduce the notation for object contours that will be used throughout of this work. We denote the contour of an object that occupies the non-empty connected region $\Omega \subset \mathbb{R}^2$ in the image by the boundary $\partial\Omega$. Further, we assume that each contour may be approximated by a polygonal chain of M vertices $\mathbf{y}_i \in \mathbb{R}^2$ encoded in a vector with dimension $N = 2M$:

$$\mathbf{y} = (\mathbf{y}_1^T, \ldots, \mathbf{y}_M^T)^T \in \mathbb{Y} \subseteq \mathbb{R}^N.$$

III. Problem Formulation

The objective of our system is to infer the objects' kinematic pose and spatial extents from camera images. Our proposed method for solving this problem combines DNNs with recursive filtering in the world frame and is outlined in Fig. 1. In essence, it relies on comparing the measured and the expected spatial expansion of the objects within the image frame. Particularly, the spatial expansion is captured in the form of the objects' silhouettes.

In this work, we examine axis-aligned bounding boxes and segmentation masks extracted by a DNN as the image observations. We further condense the large-scale image mask to the discretized contour representation using supplementary computer vision methods. Accordingly, we propose two versions for the measurement model. The objective is to detect and analyze adverse effects on the system performance upon adopting the much simpler box measurements which we consider to be the minimum manifestation of an axis-aligned object contour determined by only two framing points. Furthermore, we want to examine what benefits can be gained when increasing the amount of information and the complexity of the method using the more intricate object contours provided by the segmentation output.

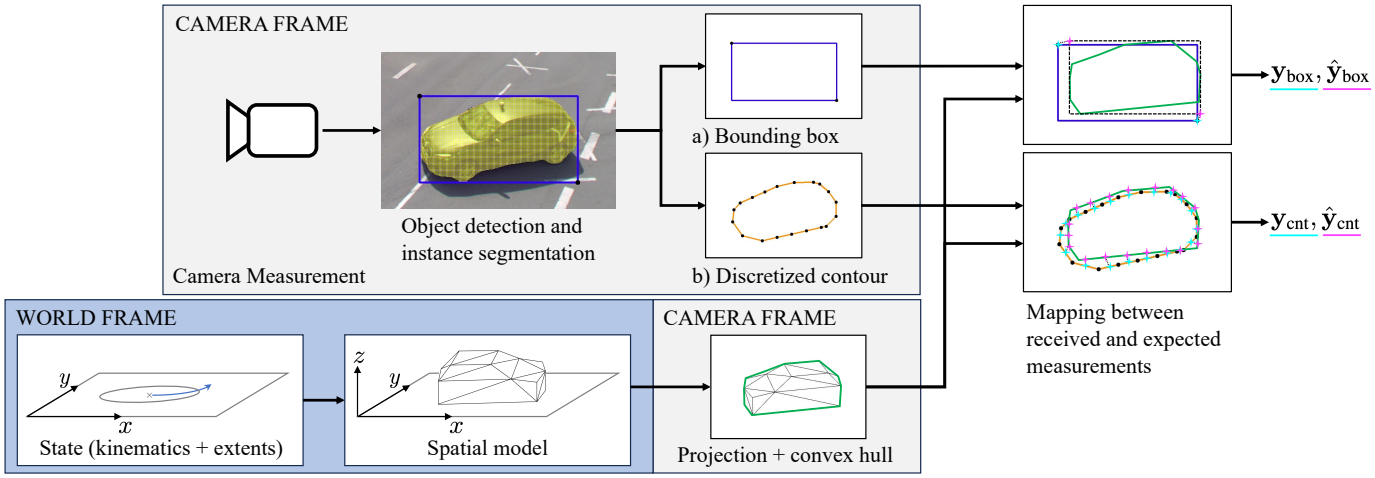


Fig. 1: Illustration of the proposed method which combines deep-learning for extracting the observation with filtering of the object state in the world frame. Two measurement models are presented: one tailored to bounding boxes (a) and one to discretized contours (b). Both versions provide a measurement $\mathbf{y}$ and an expected measurement $\hat{\mathbf{y}}$ containing point correspondences.

A. Measurement Description

We employ a deep neural network for detecting the vehicles within the RGB-encoded camera images. More specifically, the pretrained YOLOv7 model [18] is chosen because of its good performance in terms of speed and accuracy. Each detection is described by an axis-aligned bounding box specified by its top left corner $\mathbf{y}_{\text{min}} = (u_{min}, v_{min})^T \in \mathbb{N}^2$ and bottom right corner $\mathbf{y}_{\text{max}} = (u_{max}, v_{max})^T \in \mathbb{N}^2$ within the camera image. Furthermore, the YOLOv7 detection network provides a binary segmentation mask $B\colon \mathbb{N}^2 \mapsto \{0, 1\}$ which specifies for each pixel in the image if the detected object is located there or not.

1) Box Measurement Preparation: Using the top left and and the bottom right corner of the bounding box, we define a box measurement as $\mathbf{y}_{\text{box}} = (\mathbf{y}_{\text{min}}^T, \mathbf{y}_{\text{max}}^T)^T$.

2) Contour Measurement Preparation: To generate a discretized representation of the contour, we transform the binary mask B to a polygon given by the vertex chain $\mathbf{y}_{\text{cnt}}$. For this purpose, the contours are extracted from the binary image following the approach presented in [19]. Furthermore, their vertices are sampled down using a combination of a line simplification algorithm, i.e, the Douglas-Peucker method [20], and by computing the convex hull of the resulting polygon. After the simplification, excessively long segments get subdivided to ensure an approximate equidistant distribution of the vertices. The presented approach is not the only method for obtaining a discretized contour. Other suitable methods can be used here provided the result is a simplified and nearly equidistant vertex chain. Note that the number of M points that describe the complete observation varies between the time steps.

IV. State Description and Measurement Model

In the following, we present the details of the object state and its relationship to the measurements.

A. Object State

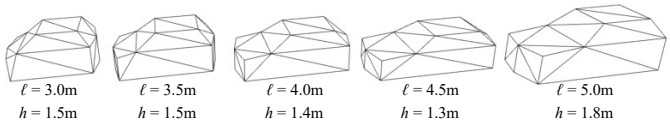
The object state is described by the vector

$$\mathbf{x} = (x, y, v, \phi, \omega, \ell, h) \in \mathbb{X} \subseteq \mathbb{R}^7.$$

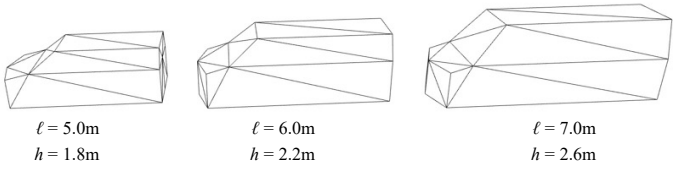
The coordinates x, y indicate the center of the object on the two-dimensional plane representing the road surface. The object velocity is described in its polar form, with the longitudinal speed v and the heading angle ϕ. The heading angle indicates the direction of the vehicle's velocity vector, as well as the rotation of the vehicle's body around its vertical axis at the center point. The change in the heading is given by the turn rate ω. We employ the Constant Turn (CT) motion model with polar velocity [21] to describe the anticipated motion patterns of the vehicles. Since we want to estimate the vehicle's extents, we include the length ℓ and the height h of the vehicle body in the state vector, whereas the width is fixed to a particular value based on the object category. We assume that all objects belong to either the *car* or the *van* category and for this classification to be known. In particular, the width of *cars* is assumed to be $1.8\,\text{m}$ and of *vans* it is $2\,\text{m}$. While object detection networks provide assessments on the object categorization, future work will need to address the uncertainty about the object classification and explore a principled approach to the stemming challenges.

B. Spatial Object Model

We approximate a vehicle's three-dimensional surface by picking a discrete set of d functions $\mathbf{q}_i\colon \mathbb{R}^2 \mapsto \mathbb{R}^3$, $i = 1, \ldots, d$ which return the vertices on the vehicle surface relative to its center, parameterized by the length ℓ and height h. Let $R_z(\phi) \in \mathbb{R}^{3\times3}$ be the rotation matrix representing the heading rotation around the z-axis. With these definitions, the approximation of the spatial expansion $O(\mathbf{x})$ of an object is



(a) Cars with width fixed to $w = 1.8\,\mathrm{m}$



(b) Vans with width fixed to $w = 2.0\,\mathrm{m}$

Fig. 2: Manifestation of the model for exemplary vehicle extents.

obtained as the set of all d state-dependent surface vertices $\mathbf{r}_i \colon \mathbb{X} \mapsto \mathbb{R}^3$, $i = 1, \ldots, d$, via

$$\begin{aligned} O(\mathbf{x}) &= \{\mathbf{r}_1(\mathbf{x}), \ldots, \mathbf{r}_d(\mathbf{x})\}, \\ \text{with } \mathbf{r}_i(\mathbf{x}) &= (x, y, 0)^T + R_z(\phi)\mathbf{q}_i(\ell, h). \end{aligned} \tag{1}$$

Choosing a simple cuboid for the spatial object model, would disregard important details of a car's actual shape, notably the lower height of the hood. Thus, we define a more intricate model that incorporates this detail. Our vehicle model uses $d = 18$ vertices whose relative placement from the center is dependent upon the estimated length and height of the object. We define two distinct models covering the categories *car* and *van*. In Fig. 2, the manifestation of the model for a few exemplary vehicle extents are shown for both vehicle categories.

C. Expected Contour Computation

Finally, the steps to compute the expected contour of an object at the particular state $\mathbf{x}$ are as follows:

1) Construct the spatial 3D model $O(\mathbf{x})$ conditioned on the object state $\mathbf{x}$ as described in Section IV-B.
2) Project the vertices of the model to the image space with
$$\mathbf{P}_\mathbf{x} \coloneqq p(O(\mathbf{x})) \subset \mathbb{R}^2.$$
The projection function $p \colon \mathbb{R}^3 \mapsto \mathbb{R}^2$ formulates the transformation from Cartesian world to image coordinates assuming a standard projective camera model provided that the calibration parameters are known.
3) Compute the convex hull of the finite point set $\mathbf{P}_\mathbf{x}$, obtaining a convex polygon $\Omega_\mathbf{x} \subset \mathbb{R}^2$ which represents the region in the image that is expected to be occupied by the object in the image space. The expected contour is determined as the boundary $\partial\Omega_\mathbf{x}$.

V. Mapping Between Received and Expected Measurements

All points on the expected object contour $\mathbf{z} \in \partial\Omega_\mathbf{x}$ represent potential sources for the points $\mathbf{y}_i$ of the discretized contour measurement. In particular, each measured point $\mathbf{y}_i$ arises from a single measurement source $\mathbf{z}_i \in \mathbb{R}^2$ and is assumed to be normally distributed around it:

$$\mathbf{y}_i \sim \mathcal{N}(\mathbf{z}_i, R_y),$$

with the noise covariance $R_y \in \mathbb{R}^{2\times2}$ for a single point. However, it is unknown which of the measurement sources are actually observed in the contour measurement. Thus, we define a model that approximates the likely mapping between the measured points and their respective source $\theta_\mathbf{x} \colon \mathbb{R}^2 \mapsto \mathbb{R}^2$. Evaluating the mapping model for all points in the contour measurement, returns an ordered list of expected measurement sources encoded in the vector

$$\hat{\mathbf{y}} = (\theta_\mathbf{x}(\mathbf{y}_1)^T, \ldots, \theta_\mathbf{x}(\mathbf{y}_M)^T)^T \in \mathbb{Y}.$$

A. Box Measurement Model

To find the correspondences between the measured and the expected axis-aligned bounding box, we need to find the extreme coordinates of all points of the expected contour $\mathbf{z} = (u_z, v_z) \in \partial\Omega_\mathbf{x}$:

$$\begin{aligned} \theta_{\text{box},\mathbf{x}}(\mathbf{y}_{\text{min}}) &= \left(\min_{\mathbf{z}\in\partial\Omega_\mathbf{x}} u_z,\ \min_{\mathbf{z}\in\partial\Omega_\mathbf{x}} v_z\right)^T, \\ \theta_{\text{box},\mathbf{x}}(\mathbf{y}_{\text{max}}) &= \left(\max_{\mathbf{z}\in\partial\Omega_\mathbf{x}} u_z,\ \max_{\mathbf{z}\in\partial\Omega_\mathbf{x}} v_z\right)^T. \end{aligned} \tag{2}$$

B. Contour Measurement Model

In order to find the corresponding point of the measured contour, we determine its closest point on the expected contour while additionally examining the contour normals. We denote the directed line segments of the polygon $\mathbf{y}_\text{cnt}$ with M vertices by

$$\mathbf{s}_i = \mathbf{y}_{(i \bmod M)+1} - \mathbf{y}_i = (u_{s,i}, v_{s,i})^T \quad \text{with} \quad i = 1, \ldots, M.$$

We replace the measurement with the center points of the line segments

$$\mathbf{y}'_\text{cnt} = (\mathbf{c}_1^T, \ldots, \mathbf{c}_M^T)^T \quad \text{with} \quad \mathbf{c}_i = (\mathbf{y}_i + 0.5 \cdot \mathbf{s}_i)^T. \tag{3}$$

Assuming a counter-clockwise vertex order, we define the outward directed normal as $\mathbf{n}((u, v)^T) = (-v, u)^T$. Let $\mathbf{z}_\perp$ be the orthogonal outward direction of the boundary point $\mathbf{z}$. The mapping function is then defined as

$$\theta_{\text{cnt},\mathbf{x}}(\mathbf{c}_i) = \underset{\mathbf{z}\in\partial\Omega_\mathbf{x},\ \mathbf{z}_\perp\cdot\mathbf{n}(\mathbf{s}_i)>0}{\arg\min} \|\mathbf{c}_i - \mathbf{z}\|. \tag{4}$$

By restricting the search space to only similarly oriented polygon segments of the expected contour, we increase the robustness of the model against wrong mappings between opposing sides.

C. Field of View Transition

Furthermore, we need to ensure a smooth transition at the boundaries of a camera's field of view (FoV). In particular, we explicitly need to handle cases when the object gets truncated at the image border to prevent estimation errors. Since the truncated measurements do not reflect the complete object, we also restrict the region $\Omega_\mathbf{x}$ that is expected to be occupied by the object to the image area given by I:

$$\Omega^*_\mathbf{x} = \Omega_\mathbf{x} \cap I.$$

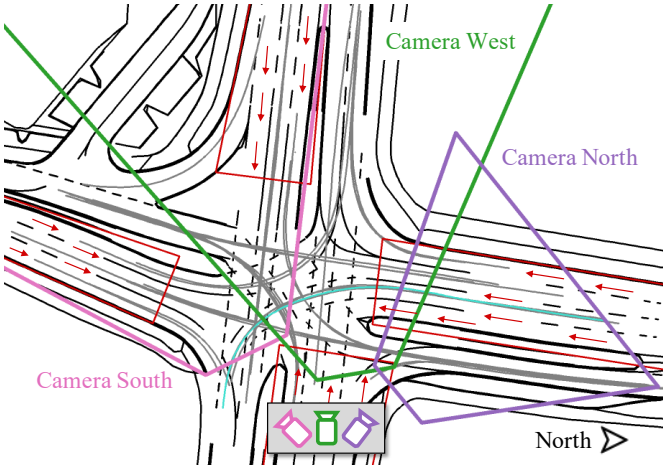


Fig. 3: Intersection layout. The field of views of the three cameras are outlined and the initialization heading angle is indicated by arrows for the approaching lanes. The examined trajectories are plotted in grey and the example trajectory analyzed in Section VI-C1 is highlighted in turquoise.

Consequently, the mapping models (2), (4) consider the restricted contour $\partial\Omega^*_{\mathbf{x}}$ instead of $\partial\Omega_{\mathbf{x}}$.

Moreover, we want to exclude the contour segments from the contour measurements that are directly located at the image border. They do not represent a valid measurement of the object outline but still enter into the update with presumably detrimental effects. We consider a segment $\mathbf{s}_i$ as invalid whose center point $\mathbf{c}_i$ is less than 10 pixels away from one of the image borders and additionally has at most an angular offset of $8°$ between their normal orientation $\mathbf{n}(\mathbf{s}_i)$ and the outward orientation of the respective image border. We exclude the particular segments from the measurement by removing their center points $\mathbf{c}_i$ from the measurement vector $\mathbf{y}'_{\text{cnt}}$ (3). Accordingly, there are no correspondences between the excluded points and the expected contour established.

D. Multi-Camera Fusion

We consider a system with multiple adjoining cameras whose FoVs might partly overlap as the particular application for our method. A single filter is employed to estimate the system state of the object by fusing the data of the multiple cameras in the world frame. It sequentially updates the object's state with the measurements of the individual cameras in the order of their recording. The available camera data is thus alternately incorporated into the global object state following the measurement-to-track fusion principle. The previously presented method to handle truncated measurements at the image borders contributes to a seamless transition between the FoVs of the adjoining cameras as the object passes through our system.

VI. Evaluation

In this section, we present experimental evaluations to assess the effectiveness of our approach. As a proof of concept, we conduct experiments with real-world data designed to investigate the localization accuracy of our system. We evaluate the method based on the bounding box measurements as described in Section V-A denoted hereafter as M1 and the method using the segmentation masks from Section V-B identified with M2.

A. Dataset

The intersection as visualized in Fig. 3 serves as the experimental setup. We consider three cameras mounted to the eastern gantry bridge as the available sensors for our experiment that cover the road with partly overlapping FoVs. They provide image streams at approximately 12 Hz. In total, we recorded five datasets by equipping different vehicles with a high-precision Global Navigation Satellite System (GNSS) device[1] and recording multiple trajectories at the intersection. Table I provides an overview of the datasets including the vehicle types. In order to obtain reference values at the exact time points as the system output, we linearly interpolate between the recorded ground truth positions. We further use consecutive ground truth positions to compute a reference heading for evaluating the estimated object heading. The true object extents are known to us and listed in Table I. On top of computing the errors between the ground truth data and the estimated output, we calculate the Wasserstein distance (WSD) proposed in [22]. We use the four object corners of the ground plane rectangle constructed with the estimated object state values and the ground truth data, respectively, for computing the WSD.

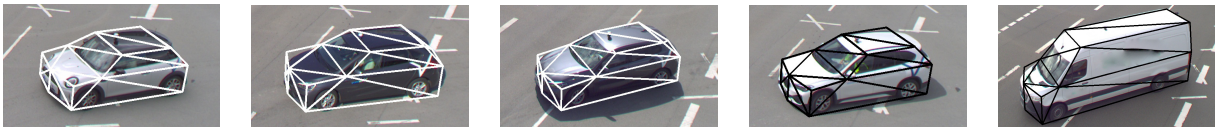
B. Implementation and Evaluation Details

In all experiments, we set the size evolution to $\sigma_\ell = 0.1\,\mathrm{m\,s^{-1}}$ and $\sigma_h = 0.5\sigma_\ell$ and the measurement noise covariance to $R_y = \mathrm{diag}(\sigma_y^2, \sigma_y^2)$ with $\sigma_y = 10$. In addition, for choosing the optimal motion model parameters, we perform a grid search by optimizing the WSD for the single left turn trajectory starting from the north contained in dataset 4 for both respective algorithms. The parameter search space is set to the intervals $\sigma_{\dot{v}} \in [1, 4]\mathrm{m\,s^{-2}}$ with step size 0.5 and to $\sigma_{\dot{\omega}} \in [0.2, 1.0]\mathrm{rad\,s^{-2}}$ with step size 0.2. We obtain the values $\sigma_{\dot{v},\text{box}} = 3\,\mathrm{m\,s^{-2}}$ and $\sigma_{\dot{v},\text{cnt}} = 4\,\mathrm{m\,s^{-2}}$ as the optimal acceleration noise for the two methods and a shared optimal angular acceleration noise of $\sigma_{\dot{\omega}} = 0.2\,\mathrm{rad\,s^{-2}}$.

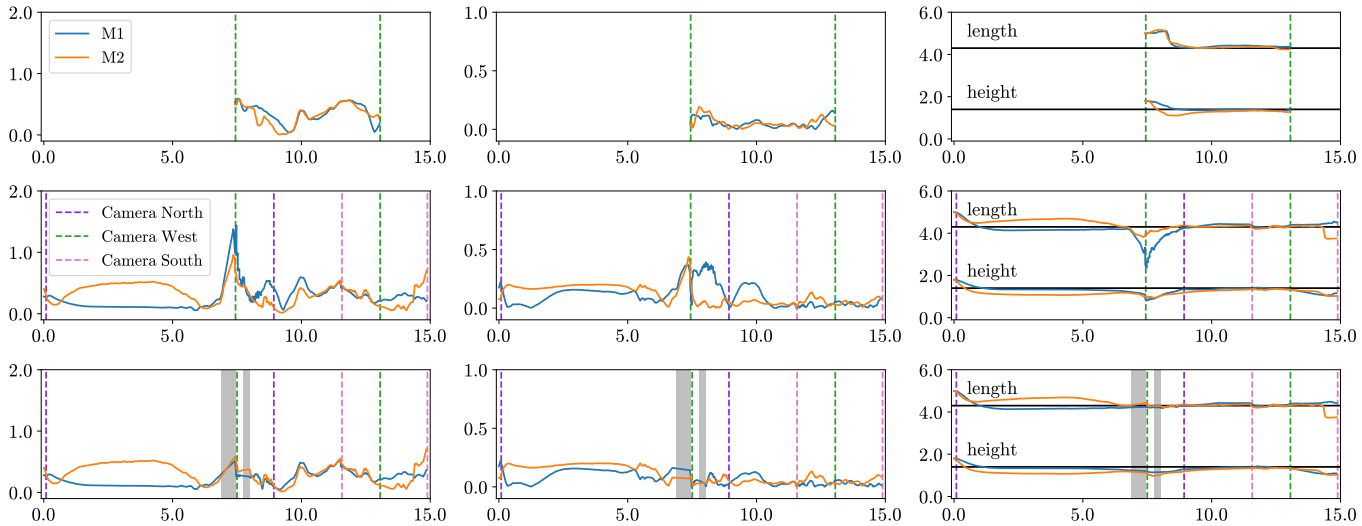
The system state is initialized with the first available measurement. Independent of the measurement model, we choose the center of the lower bounding box half as the start position. The corresponding position on the road plane is determined using the inverse of the projection function p. We take advantage of the available context information given by the road intersection layout and we set the start heading to the approximate direction of the respective road lane the object is approaching the intersection from (see Fig. 3). The initial extents for cars are set to $\ell = 5\,\mathrm{m}$ and $h = 1.8\,\mathrm{m}$ and for vans to $\ell = 6\,\mathrm{m}$ and $h = 2\,\mathrm{m}$, respectively.

[1]Emlid Reach RS2 using Real-Time Kinematic (RTK) correction provided by the real-time positioning service SAPOS resulting in centimeter-level precision.

TABLE I: Overview on the recorded datasets accompanied by example images with visualized spatial models for exemplary filter output. A combined total of 39 trajectories are examined.



	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5
Type	MINI Cooper S	BMW 2 Active	BMW 1	BMW X1	Mercedes Sprinter
Length [m]	3.85	4.386	4.3	4.447	6.97
Height [m]	1.414	1.576	1.4	1.598	2.62
Category	*Car*	*Car*	*Car*	*Car*	*Van*



(a) Position error [m] (b) Heading error [rad] (c) Estimated extent [m]

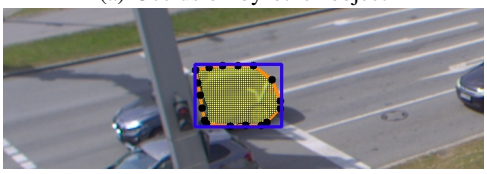
Fig. 4: Results for the example trajectory from dataset 3 over time [s] for the single-camera (top), the multi-camera experiment (middle) and the multi-camera experiment excluding the measurements at the time points marked in grey (bottom). The time point of the first and the last measurement of the respective cameras are indicated with dashed lines. The true extent values are shown as horizontal black lines.

C. Results and Discussion

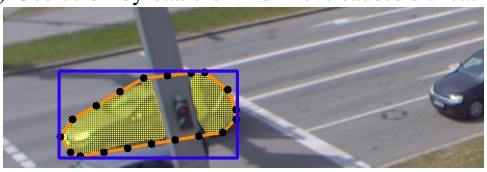
1) Example trajectory: For an in-depth analysis of the methods' behavior, we select the single left turn trajectory starting from the north contained in dataset 3 (see Fig. 3). First, we consider only the data from the West camera, to analyze the methods' performance for the single-camera case. The results are shown in the first row of Fig. 4. In terms of position and heading error, both methods provide similar outputs for the data of camera West. However, we can see that while both quickly correct the inital extent values to more accurate values, method M2 suffers from a noticeable underestimation of the height. The results for the multi-camera case are presented in the middle row of Fig. 4 and depict significantly larger error peaks in the plots. Reviewing the data of the recording, we find that our test vehicle was partly occluded by another vehicle for the first 7 s while waiting at the traffic lights (see example image in Fig. 5a). Then, after it started to traverse the intersection, it was occluded by the traffic light pole which temporarily caused a truncation of the measurements (see Fig. 5b). Both occlusion scenes happen in the data of the camera North and thus, have no deteriorating effect on the previously investigated single-camera case. The first method M1 outperforms the second method M2 in the first occlusion scenario. Looking at Fig. 5a, we can notice that the bounding box still encloses our vehicle relatively well in this particular occlusion situation. However, the contour which represents the down sampled convex hull of the segmentation mask, causes an erroneous estimated state when fitting the object to the truncated silhouette. On top of the larger localization and heading errors for method M2 in the first 7 s, we also overestimate the length and underestimate the height.

The second occlusion scenario caused by the traffic light pole produces the major error spikes right before the object enters the FoV of the West camera. Especially in the case of M1, the state estimation process gets massively destabilized by the truncated bounding boxes. We notice a severe peak in the position error, a subsequent increase in the heading errors and strong deviations in the estimated extent, especially in the object length. Method M2 exhibits more resilience against this error source because the introduced errors in the plots are less pronounced and are also compensated faster. One important point to add here is that taking the convex hull of

(a) Occlusion by other object



(b) Occlusion by static environment causes truncation



(c) Occlusion by static environment

Fig. 5: Three example images of the North camera showing the occlusion scenes taken from the left-turn trajectory in dataset 3. The measurement data is visualized: the bounding box (blue), the segmentation mask (yellow) and the vertices (black) and edges (orange) of the simplified contour.

the segmentation mask ensures that we obtain a single, closed contour polygon despite any discontinuities in the mask as visible in the example image Fig. 5c.

In order to differentiate the effect of the occlusion by the pole on the system output, we conduct the experiment a third time while excluding the particular heavy truncated measurements. We present the results in the bottom row of Fig. 4 and highlight the time points of the 11 excluded measurements in grey. The deterioration caused by the first occlusion scenario is still visible, however, the subsequent spikes right before the FoV border of camera West are gone. In general, the transition between the cameras' FoVs do not introduce major errors and on top of that, we can also observe a slight reduction of the position error in the intervals where the data of two cameras are fused compared with the single-camera case in the top row. In summary, we draw the conclusion, that both methods need to be complemented by a dedicated approach to handle partial occlusion scenarios caused by dynamic objects and by the static environment.

2) Performance statistics: In addition, we provide the evaluation results in the form of violin plots in Fig. 6 showing the distribution of the computed state errors and the WSD for all datasets collectively with a combined total of 39 trajectories. From Fig. 6g, the first thing to note is that the WSD metric shows better performance for M2 compared to M1, both on average and given by the 95th percentile values. The average localization errors of both methods (see Fig. 6a) are with $0.385\,\mathrm{m}$ and $0.342\,\mathrm{m}$ in a very reasonable, but also a very comparable range. In turn, the heading errors are in general considerably larger for M1 compared to M2 and, particularly, the mean heading error is with $0.225\,\mathrm{rad}$ more than four times larger than the mean of M2 with $0.055\,\mathrm{rad}$. This is not a surprising outcome given that the discretized contour captures the object's orientation more explicitly than the bounding box. From Figs. 6e and 6f, we can conclude that both methods, but especially M2, in general tend to underestimate the object extents. Moreover, in all plots, we can see that the extreme values of M1 are significantly larger than the ones of M2. Again, we deduce a higher robustness of the method M2 against erroneous measurements that have a greater destabilizing effect on M1.

To summarize the conducted evaluations, both methods offer on average and in most cases convincing results for the localization and the pose and extent estimation of the object. While the box measurement model has the advantage of being considerably more simple, it demonstrated a lower robustness against severe measurement errors in the form of truncated bounding boxes. On the other hand, the contour measurement model includes a lot more computation steps. But it notably lowered the heading errors and displayed significantly reduced error peaks. Furthermore, occlusion scenarios are found to have a substantial negative effect on the system performance.

VII. Conclusion

In this paper, we presented an approach for estimating the pose and extents of vehicles based on the data from a multi-camera surveillance system. It combines deep neural networks with recursive filtering in the world frame. We proposed two measurement models: one tailored to bounding boxes and one to segmentation masks. Our evaluations with real-world data from an intersection proved that our methodology delivers convincing results for the localization and the pose and extent estimation. The comparison of the two measurement models has shown that the more complex contour model did provide a performance increase, in particular concerning the estimated heading. Moreover, it demonstrated a considerably higher robustness against severe measurement errors. Despite the promising results, we have identified weaknesses and issues that need to be addressed. In particular, both methods need to be complemented by a dedicated approach to handle partial occlusion scenarios. Furthermore, the system needs to be extended to cover other traffic participants and the algorithm must be integrated into multi-object tracking algorithms in order to provide genuine traffic monitoring capabilities.

References

[1] J. G. López, A. Agudo, and F. Moreno-Noguer, "Vehicle pose estimation via regression of semantic points of interest," in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2019, pp. 209–214.

[2] L. Ke, S. Li, Y. Sun, Y.-W. Tai, and C.-K. Tang, "GSNet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision," in *Computer Vision–ECCV 2020*, 2020, pp. 515–532.

[3] S. Li, Z. Yan, H. Li, and K.-T. Cheng, "Exploring intermediate representation for monocular vehicle pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1873–1883.

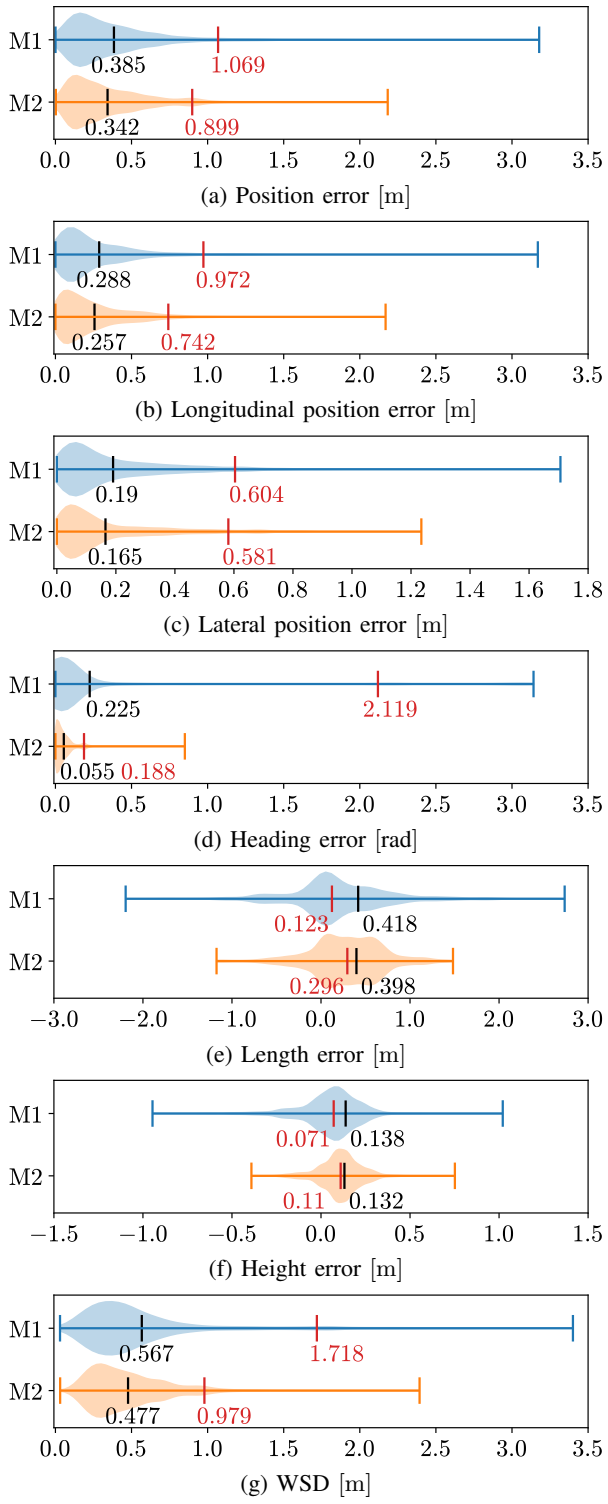


Fig. 6: Distribution of the WSD metric and the position, heading and extents errors for all individual time steps of the examined trajectories for all datasets collectively. The mean value is shown in black and the value of the 95th percentile in red. In Figs. 6e and 6f, we keep the sign of the errors (negative values signify overestimation, positive values underestimation), but compute the mean based on the absolute values. We further provide the median here instead of the 95th percentile.

[4] L. Yang, K. Yu, T. Tang, J. Li, K. Yuan, L. Wang, X. Zhang, and P. Chen, "BEVHeight: A robust framework for vision-based roadside 3D object detection," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 21 611–21 620.

[5] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *2017 IEEE international conference on robotics and automation (ICRA)*, 2017, pp. 2011–2018.

[6] K. Schmeckpeper, P. Osteen, Y. Wang, G. Pavlakos, K. Chaney, W. Jordan, X. Zhou, K. Derpanis, and K. Daniilidis, "Semantic keypoint-based pose estimation from single RGB frames," *Field Robotics*, vol. 2, no. 1, pp. 147–171, 2022.

[7] J. Shi, H. Yang, and L. Carlone, "Optimal and robust category-level perception: Object pose and shape estimation from 2-D and 3-D semantic keypoints," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 4131–4151, 2023.

[8] M. Coenen and F. Rottensteiner, "Pose estimation and 3D reconstruction of vehicles from stereo-images using a subcategory-aware shape prior," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 181, pp. 27–47, 2021.

[9] R. Wang, N. Yang, J. Stückler, and D. Cremers, "DirectShape: Direct photometric alignment of shape priors for visual vehicle pose and shape estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 11 067–11 073.

[10] A. Scheel, F. Gritschneder, S. Reuter, and K. Dietmayer, "Fusing radar and scene labling data for multi-object vehicle tracking," in *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, 2017, pp. 11–20.

[11] N. A. Piga, F. Bottarel, C. Fantacci, G. Vezzani, U. Pattacini, and L. Natale, "MaskUKF: An instance segmentation aided unscented Kalman filter for 6D object pose and velocity tracking," *Frontiers in Robotics and AI*, vol. 8, 2021.

[12] C. Creß, W. Zimmer, L. Strand, M. Fortkord, S. Dai, V. Lakshminarasimhan, and A. Knoll, "A9-Dataset: Multi-sensor infrastructure-based dataset for mobility research," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 965–970.

[13] L. Strand, J. Honer, and A. Knoll, "Systematic error source analysis of a real-world multi-camera traffic surveillance system," in *2022 25th International Conference on Information Fusion (FUSION)*, 2022, pp. 1–8.

[14] L. Strand, J. Honer, and A. Knoll, "Modeling inter-vehicle occlusion scenarios in multi-camera traffic surveillance systems," in *2023 26th International Conference on Information Fusion (FUSION)*, 2023, pp. 1–8.

[15] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[16] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, 2000, pp. 153–158.

[17] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[18] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 7464–7475.

[19] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[20] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[21] X. Rong Li and V. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[22] S. Yang, M. Baum, and K. Granström, "Metrics for performance evaluation of elliptic extended object tracking methods," in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2016, pp. 523–528.