Multi-Modal Identification and Tracking of Vehicles in Partially Observed Environments

Daniel Becker[†], Alexander Binder[*§], Jens Einsiedler[‡*], Ilja Radusch[*]

[*] Fraunhofer Institut for Open Communication Technologies (FOKUS), Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
{alexander.binder,ilja.radusch}@fokus.fraunhofer.de
[†] Daimler Center for Automotive Information Technology Innovations (DCAITI), Ernst-Reuter-Platz 7, 10587 Berlin, Germany
daniel.becker@dcaiti.com
[‡] Fraunhofer Application Center for Wireless Sensor Systems, Am Hofbräuhaus 1, 96450 Coburg, Germany
jens.einsiedler@iis.fraunhofer.de
[§] Machine Learning Group, TU Berlin, Marchstr. 23, 10587 Berlin, Germany

***Abstract*—High-quality positioning is of fundamental importance for an increasing variety of advanced driver assistance systems. GNSS-based systems are predominant outdoors but usually fail in enclosed areas where a direct line-of-sight to satellites is unavailable. For those scenarios, external infrastructure-based positioning systems are a promising alternative. However, external position detections have no identity information as they may belong to any object, i.e. they are anonymous. Moreover, the area covered by external sensors may contain gaps where objects cannot be observed leading to a correspondence problem between multiple detections and actual objects.**

We present a global tracking-by-identification approach as extension to existing local trackers that uses odometry sensor data of vehicles to find the corresponding subset of external detections. Thus, our approach enables the assignment of anonymous external detections to a specific vehicular endpoint and the estimation of its current position without requiring an initial location.

The problem is decomposed resulting in a two step approach. The first algorithm determines possible track segment combinations which are used as track hypotheses. The track hypothesis generation algorithm considers spatio-temporal relationships between track segments, thus avoiding exponentially growing complexity inherent to data association problems. The second algorithm compares track hypotheses to the relative vehicle trajectory using pseudo-distance correlation metrics.

In a detailed evaluation, we demonstrate that the proposed approach is able to reliably perform global tracking and identification of camera-observed vehicles in real-time, despite relatively large coverage gaps of the external sensors.

I. Introduction

In outdoor areas, global navigation satellite systems (GNSS) such as global positioning system (GPS) or Galileo are the predominant positioning technology. However, in enclosed areas and even urban environments where the line-of-sight to satellites is interrupted, GNSS-based systems suffer from a highly degraded performance or become completely inoperable [1], [2]. Nevertheless, modern vehicles are equipped with a rapidly growing number of advanced driver assistance systems. Above all, autonomous driving [3] depends on highly accurate positioning information, especially in challenging urban areas such as urban canyons or underground carparks. Consequently, the need for alternative positioning technologies has arisen. Unlike outdoors where one dominant technology has been established, many different approaches have been proposed for indoor positioning [4]. Generally, positioning systems can be classified according to two criteria: *Order* and *Perspective*.

The first classification *Order* refers to the mathematical order of the time derivative of the quantities measured by the positioning system. We will refer to systems measuring the time derivative of order zero as absolute and systems measuring time derivatives of order larger than zero as relative. For instance, a positioning system which measures the position of a vehicle on a road $x(t)$ (e.g. GPS) is classified as absolute. On the other hand, a system which measures the velocity $\dot{x}(t)$ (e.g. wheel encoder) or acceleration (e.g. inertial measurement unit (IMU)) $\ddot{x}(t)$ is classified as relative. This classification is meaningful as relative positioning systems alone are insufficient without an initial position, i.e. $x(t)$ cannot be inferred from $\dot{x}(t)$ without an initial position $x(0)$ (cf. [5]). The second classification *Perspective* differentiates sensor readings acquired from an internal or external perspective. Thereby, internal means that an object positions itself according to its environment. Examples for internal positioning systems are WiFi fingerprinting [6], magneto techniques [7], odometry [8] and IMUs [9]. On the other hand, external approaches determine the state of the object from an external perspective, i.e. the object which needs position information (e.g. pedestrian, vehicle) is unaware of the detection process. Also, the identity of externally-observed objects is often unknown, i.e. external position measurements are anonymous. Hence, in order to assign and transmit anonymous external position detections to the correlated object, an additional identification step is required. Examples are infrastructure visual positioning (e.g. [10], [11], etc.), stationary laser scanners [12] and traffic speed cameras [13].

Based on this classification in the two dimensions *order* (absolute or relative) and *perspective* (internal or external), four different classes can be created, as illustrated in Tab. I.

One promising research field is visual (indoor) positioning based on infrastructure-based cameras (e.g. surveillance cameras). Above all, the utilization of existing infrastructures as positioning system is very cost-effective. Moreover, cameras

		Order	
		Absolute	*Relative*
Perspective	*Internal*	GPS [1] [2], WiFi Fingerprinting [6], etc.	Odometry [8], IMU [9]
	External	Infrastructure camera [10] [11] / LiDAR [12]	Traffic Speed Camera [13]

TABLE I
CLASSIFICATION OF POSITIONING TECHNIQUES ACCORDING TO PERSPECTIVE (INTERNAL, EXTERNAL) & ORDER (ABSOLUTE, RELATIVE).

have relatively high resolutions translating into high positioning accuracy [14]. According to Tab. I, these systems can be classified as absolute, external systems. Because of the external perspective, one of the key challenges is to provide the position information to the appropriate vehicle.

Tracking techniques can be applied to differentiate multiple objects within single and multiple camera views [15]. Thereby, each track segment has a local identity denoted by a track identifier which is independent of the global identity (e.g. vehicle number plate). However, systems with only a single camera per view are prone to identity switches in the presence of obstructions. Moreover, the fact that there are coverage gaps between the observed areas makes continuous tracking difficult. As a consequence, it is challenging to decide which track segments belong to which object over a certain time period, often referred to as *correspondence problem*.

In this work, we present a multi-modal global tracking-by-identification methodology for vehicular environments which are partially observed by external sensors. Our proposed approach is based on existing local trackers (i.e. tracking objects within one camera view) and enables the global association of anonymous detections to vehicular endpoints as well as the localization of the vehicles without requiring any initial location. To achieve this, we use internal relative sensor modalities of the vehicle (i.e. odometry information from wheel and steering encoders) to find the subset of external detections corresponding to the vehicle's relative trajectory. More specifically, we first present an algorithm that combines multiple individual track segments in order to generate a set of track hypotheses. Finally, we introduce pseudo-distance correlation metrics for measuring the similarity between the vehicle's relative sensor data and the previously generated track hypotheses.

The paper is organized as follows. In Section 2, a brief overview of related work is given. A detailed investigation of the problem and prerequisites is given in Section 3. The methodology is described in Section 4. In Section 5, we present the experimental evaluation conducted in our underground carpark test site. A conclusion and future outlook follow in Section 6.

II. RELATED WORK

There are numerous approaches that merge external visual positioning systems with sensors at the target, which essentially represents a fusion of internal and external positioning systems. The majority of approaches uses a combination of external camera positioning and internal *absolute* positioning systems (e.g. WiFi, UWB, ultrasound, RFID, etc.). However, there are also some approaches based on combining external camera positioning with internal *relative* positioning systems (e.g. IMU, gyro, odometry, etc.).

The first group of techniques is more straightforward as two sets of absolute positions can be compared directly (e.g. Euclidean distance between 1 GPS position and N camera-observed positions). There are a number of works falling into this group. In [16], infrastructure video cameras are used to detect pedestrians and WiFi positioning is employed on the user's smartphone which in combination enables the assignment of anonymous camera detections to users and thus the provisioning of accurate positioning data from the cameras. A similar approach is presented in [17], which fuses camera and radio (UWB) infrastructure to locate pedestrians equipped with UWB sensors.

Regarding the second group of techniques, there is the benefit of being more economic as the sensors are usually cheaper (e.g. IMU) and do not have to rely on any infrastructure (e.g. WiFi). But the downside of these techniques is an increased complexity, as 1) detected objects need to be on the move (relative sensors only notice the change of the absolute state), 2) there is a complex relationship between camera-observed and sensor-measured movement which needs to be modelled, 3) measurement errors of relative sensors grow unbounded over traveled distance (especially for IMUs where double integration is performed: acceleration-speed-position), 4) these methods fail if target movement is synchronized (which happens in crowded areas e.g. subway stations where people walk in close formations), 5) complexity grows exponentially in worst case (considering all possible combinations between N sensors and M camera-observed tracks is $O(N^{M!})$ [18]).

Despite these challenges, there are some publications falling into this group. An innovative approach for the fusion of an infrastructure-based camera positioning system with belt-mounted accelerometer sensors is presented in [18]. They proposed a correlation metric for comparing the characteristic up-and-down movements of pedestrians in the camera observed area with the measurements of the accelerometer. Another interesting approach proposed in [19] uses waist-mounted accelerometers to estimate the person's kinetic energy and compare it to the movement silhouettes in the motion history image (MHI) of the camera positioning system.

All in all, there is only a small number of research work dealing with the fusion of external *absolute* and internal *relative* positioning systems with the goal of localizing *pedestrians* but to the best of our knowledge there are no approaches doing this for *vehicles*, especially in enclosed structures such as underground carparks.

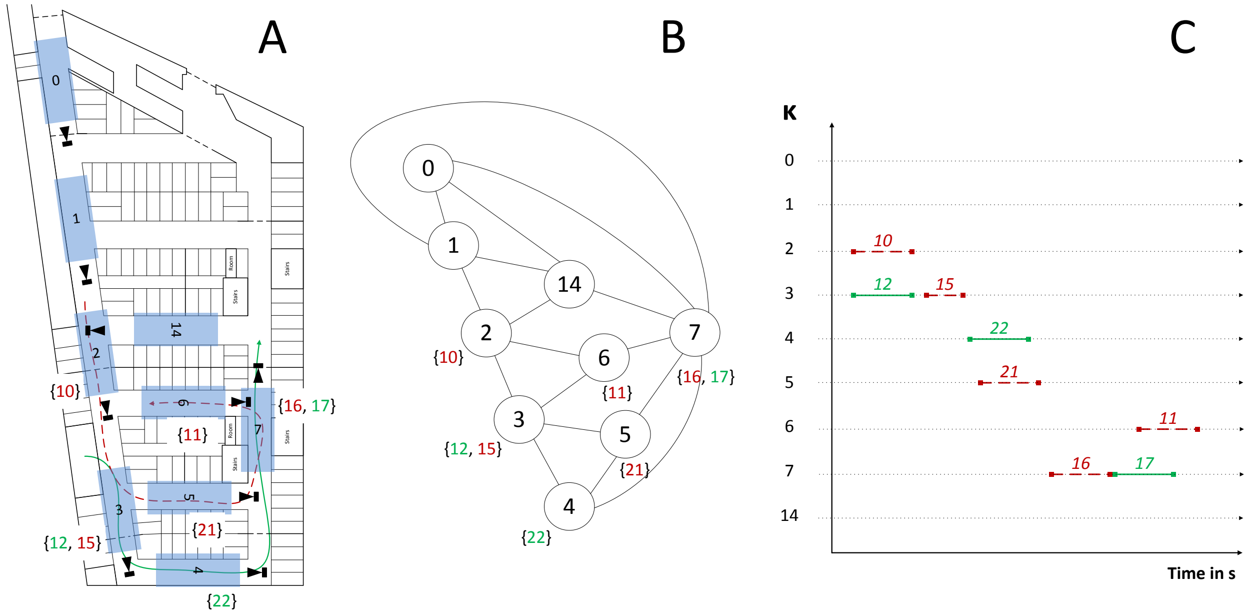

Fig. 1. Different representations of Track Segments. A Carpark map with camera-observed lane segments (includes camera identifiers κ) and trajectories of two vehicles (marked in red dashed and green straight resp.). B Graph representation of carpark lanes with identifiers of track segments per vertex. C Temporal properties of individual track segments of both vehicles.

III. Problem description

Hereinafter, we will elaborate on the problems our approach aims to solve. We assume that our proposed global tracking-by-identification technique is built on top of existing local trackers operating in non-overlapping views. Although our approach works with any type of external sensors (e.g. camera, LiDAR, binary, etc.), we focus on infrastructure cameras, which are installed in our carpark test site (cf. Fig. 1 A).

Camera-based external positioning technologies have numerous advantages: 1) High camera resolution translate into high positioning accuracies, 2) High frame rates (e.g. 30 Hz) enable tracking of relatively fast objects, 3) Image processing algorithms (e.g. in OpenCV library) are relatively mature and 4) Existing infrastructures yield cost-effective solutions.

Despite all these advantages, there are two main challenges with external camera-based positioning techniques: 1) *Unknown identity*: Detected objects are anonymous, i.e. they have no identity information. 2) *Unobserved areas*: Cameras cover the straight lanes but not the curves (cf. Fig. 1 A), resulting in *coverage gaps* where vehicles are *invisible*.

Furthermore, we assume a simple vehicle movement model for the movement of common front-steering vehicles as described by the following state dynamics [20]:

$$a(x, y, v, \psi, \omega) = \begin{pmatrix} x + v\,\Delta t \sin(\psi) \\ y + v\,\Delta t \cos(\psi) \\ v \\ \psi + \Delta t\,\omega \\ \omega \end{pmatrix} \qquad (1)$$

where x and y are coordinates on a Cartesian coordinate system relative to a fixed reference point (e.g. Universal Transverse Mercator), v is the vehicle's speed in direction of travel, ψ is the vehicle's heading direction (i.e. north=0, east=$\frac{\pi}{2}$, ..) and $\omega = \dot{\psi}$ i.e. the yaw rate of the vehicle.

Moreover, obtained detections of the visual positioning system are denoted as c_n with the following properties:

$$c_n = \{t_{cam}, x, y\} \qquad (2)$$

where t is the measurement time in seconds, x and y in Cartesian coordinates. Additionally odometry measurements o_m are obtained of vehicular sensor modalities:

$$o_m = \{t_{odo}, v, \omega\} \qquad (3)$$

where t is the measurement time in seconds, v the vehicle's velocity and ω the vehicle's yaw rate.

For typical carpark environments (cf. Fig. 1 A), the following assumptions are valid:

- Vehicle movement is constrained to narrow lane-segments
- Lane-segments are *straight* and connected by curves (angle between 70 and 110 degrees)
- Cameras observe only straight lane segments but not curves between segments
- Maximum speed of vehicles in carparks approx. $5\frac{m}{s}$
- Vehicle's movement is limited to a low degree of freedom due to front steering geometry [20], e.g. vehicles cannot suddenly change direction within a lane
- Based on the low camera angle, the detection range for each camera (= max. track length) is max. 25m

Based on these assumptions, individual positions c_n are clustered to coherent *track segments* S_i which represent the

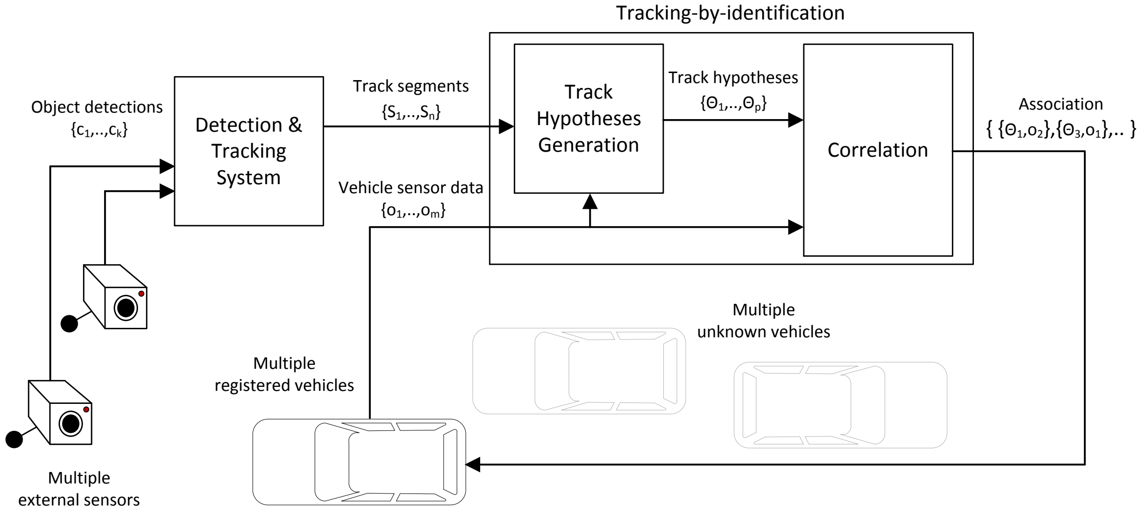

Fig. 2. System components and data flow overview: Tracking-by-identification approach receives track segments S and vehicle sensor data O, creates plausible track hypotheses Θ and returns an association of best matching O and Θ.

movement of an individual vehicle within a single view:

$$S_i = \{t_{start}, t_{end}, c_{start}, c_{end}, \{c_1, .., c_n\}, TID, \kappa\} \quad (4)$$

where t_{start} and t_{end} refers to the time of the first and last detection resp. (i.e. marking track segment *lifetime*), c_{start} and c_{end} refers to the first and last camera-detected position resp. (i.e. point difference corresponds to track segment length), $\{c_1, .., c_n\}$ is a set of all camera detections. TID denotes the local track identifier assigned by the tracker and κ is the ID of the camera which captured the detections in this track. As we assume that camera views are non-overlapping all detections in a track are captured by a single camera.

IV. Methodology

After having elaborated on the specific problems in the previous section, we will now propose a solution: First, a *generic optimization problem* is defined which represents a holistic solution. However, due to the high complexity, the problem is split into the two parts *track hypothesis generation* and *correlation metrics* described subsequently.

A. Generic optimization problem

With a given set of track segments $S = \{S_1, S_2, \ldots, S_M\}$ and a relative trajectory from odometry data O, we can formulate the track assignment problem as an optimization problem over the track hypotheses Θ obtained by performing all possible track segment combinations using a power-set $\mathcal{P}(S)$. Thereby, $|\Theta|$ denotes the number of track hypotheses. The relative trajectory O can be obtained by integrating the odometry measurements o_m from Eq. (3) according to the model in Eq. (1). Also, $|O|$ denotes the total length of the relative vehicle trajectory, referred to as *travelled distance*.

In order to define the optimization problem, we introduce some required notation. Given a track segment S_i, let O_i being the relative track segment from O which consists of those positions from O which match the detections in S_i in time. O_i is obtained by taking for each triple (x, y, t) of position and time from S_i that triple $(\hat{x}, \hat{y}, \hat{t})$ from O_i, such that $|t - \hat{t}|$ is minimal among all triples in O. Given a set of relative positions O_i, we may compute a distance function $d(S_i, O_i)$ between the positions in S_i and O_i. Also, let T be the operator which yields the set of time points from any track. Then it can be stated that a set of time points is *less in time* than another set if: $T(S_1) < T(S_2) \Leftrightarrow \max_{t \in T(S_1)} t < \min_{u \in T(S_2)} u$.

Finally, our assumptions, in particular the requirement of disjoint sensor views, allow us to represent the regions which are covered by an absolute sensor as a vertex in a graph. Two vertices in a graph are connected, if a vehicle can reach one sensor view from another sensor view without crossing a third sensor view. Figure 1 B shows this graph derived from the carpark layout given in Figure 1 A. Let then $V(S_i)$ be the graph vertex of track segment S_i and $D(S_i)$ the set of all direct neighbors of $V(S_i)$ in the graph. Hence, the optimization problem can be stated as the following:

$$\min_{\Theta \in \mathcal{P}(S)} \sum_{S_i \in \Theta} d(S_i, O_i) \quad (5)$$

$$\text{such that } \forall i \leq |\Theta| - 1 : T(S_i) < T(S_{i+1}) \quad (6)$$

$$\text{and } \forall i \leq |\Theta| - 1 : S_{i+1} \in D(S_i) \cup V(S_i) \quad (7)$$

This optimization problem encodes our desired requirements. Firstly, for each absolute track segment $S_i \in \Theta$, the spatial distance $d(S_i, O_i)$ to the relative track segment O_i should be small. The first condition, Eq. (6), enforces temporal consistency: The absolute track segments are ordered in time and are non-overlapping in time. The second condition, Eq. (7), encodes graph consistency: The next absolute track segment lies either in the same graph vertex as its predecessor S_i or it lies in a neighboring vertex.

Note that this optimization problem runs over a set of size $2^{|S|} = 2^N$. Another way to formulate the optimization problem is to consider individual camera detections c_i instead of track segments S_i. In this case, the complexity would increase exponentially with the number of camera detections. Thus, we introduce an algorithm for an approximate search for the best combination of track segments, i.e. *track hypotheses generation*, in the following.

B. Track hypotheses generation

The *track hypotheses generation* algorithm solves the first sub-problem, i.e. the selection of combinations of track segments likely to represent the trajectory of a single vehicle. The motivation is to reduce the computational complexity by reducing the number of subsequent calculations (i.e. $d(\Theta_i, O_i)$).

Given a set of track segments S_i, appropriate combinations (i.e. track hypotheses) Θ_i need to be determined and matched to the vehicle's trajectories O_i recorded by their onboard sensors. The amount of data in O_i and S_i (and also in Θ_i) depends on the time window ΔT_K, e.g. all data of the last $\Delta T_K = 30s$ is buffered and considered in the algorithm.

For a small number of S_i, it is trivial to *try out* all combinations. However, as the complexity is $O(2^N)$ (i.e. growing exponentially with N), this becomes quickly unmanageable. Thus, we propose an algorithm that generates plausible track hypotheses considering the following spatio-temporal constraints:

- Spatial neighborhood of camera views (i.e. only track segments on adjacent views are connected) Eq. (7)
- Track segments cannot have temporal overlap Eq. (6)
- Estimated vehicle speed between tracks cannot surpass threshold (e.g. 5 m/s)
- Approx. min./max. length of vehicle trajectories determines number of track segments per hypothesis

Fig. 1 A and C shows an example which illustrates two vehicle trajectories and their track segments inside the map and time view resp. In this example, each track segment is marked in the color according to the vehicle it is associated to (i.e. red/green in the figure). However in the actual system, all track segments are anonymous and the goal of this approach is to determine their correct association.

In order to enforce the spatial neighborhood constraint, a representation of a *reference graph* of all carpark lanes is created as shown in Fig. 1 B. In this case, an undirected graph representation has been derived from the lane network layout shown in Fig. 1 A. As carparks often consist of one-way streets, it is also viable to choose a directed graph representation. Directed graphs can be processed in a computationally more efficient way, since edges can be traversed in only one direction, thus limiting the total number of possible vertex combinations. For the sake of universality, we select the undirected graph representation.

```
Input  : A set of track segments $\{S_1, S_2, .., S_n\}$ and
         a set of odometry data $\{O_1, O_2, .., O_m\}$
Output: A set of track hypotheses $\{\Theta_1, \Theta_2, .., \Theta_o\}$
Scombs ← ∅;
Scombs' ← ∅;
minTravDist ← getMinTravDist($\{O_1, .., O_m\}$);
maxTravDist ← getMaxTravDist($\{O_1, .., O_m\}$);
minO ← getMinOrder(minTravDist);
maxO ← getMaxOrder(maxTravDist);
G(V,E) ← GetGref();
foreach $V$ in G(V,E) do
    if $V.\Phi$ = {} then
        G(V,E).remove(V) ;
    end
end
connV ← getConnV(G(V,E),minO,maxO);
Scombs ← getScombs(connV);
foreach $comb$ in Scombs do
    if isPlausible($comb$) then
        Scombs' ← Scombs' ∪ {$comb$};
    end
end
return Scombs'
```

Algorithm 1: Track hypotheses generation algorithm

The pseudo code of our track hypotheses generation algorithm is displayed in Alg. 1. Comparing with the component diagram in Fig. 2, the main input of the track hypotheses generation is a set of track segments $\{S_1, S_2, .., S_n\}$ and the auxiliary input is a set of vehicular sensor data $\{O_1, .., O_m\}$. The output of the algorithm is a set of track hypotheses $\{\Theta_1, \Theta_2, .., \Theta_o\}$. The single steps are as follows:

First of all, after initializing the sets *Scombs* and *Scombs'* which will hold the output result, the global min. and max. *travelled distance* of all vehicles is calculated considering *getMinTravDist()* and *getMaxTravDist()*. These functions calculate min/max($|O_1|, .., |O_m|$). Subsequently, the result (i.e. minTravDist, maxTravDist) is used to obtain the min. and max. order of the track segments per hypothesis, i.e. the number of track segments per combination (cf. *getMinOrder()* & *getMaxOrder()*). This is a simple heuristic which helps minimizing the computational burden of the subsequent combinatorical algorithms (i.e. *getConnV()* and *getScombs()*). Based on the carpark layout (cf. Fig. 1 A), the lengths of track segments are in the range of 5m to 30m. Assuming the *travelled distance* of all vehicles is below a certain threshold (e.g. 15m), it is unlikely that a vehicle has passed more than one camera view. Thus, only track hypotheses of order one need to be considered. Similarly, in the case that the *travelled distance* of all vehicles is very high (e.g. 50m), it is likely that multiple camera views have been passed, thus only hypotheses of higher order (e.g. 2 or 3) are considered.

After that, an instance of the reference graph (cf. Fig. 1 B) is stored in variable $G(V, E)$. Complying with usual graph notation [21], $G(V,E)$ consists of a set of vertices and edges, denoted as V and E respectively. However, we modified the definition of a vertex v in $V=\{v_1,..,v_p\}$:

$$v=\{\kappa,\Phi=\{S_1,..,S_i\}\} \tag{8}$$

where κ is the vertex ID corresponding to the lane identifier (cf. Fig. 1 B) and Φ is a set of track segments that is associated to the camera view represented by this graph vertex. Following up in the algorithm, all vertices are iterated and the ones where Φ is empty are removed (along with all connected edges).

Subsequently, in order to comply with condition Eq. (7) the modified graph $G(V,E)$ is used to obtain all *connected* vertex combinations of the order in the range of *minOrder* to *maxOrder* (cf. *getConnV()*). Although this step intuitively appears to be simple, it is computationally very challenging, especially for large connected graphs. The most straightforward algorithm for finding all *connected* vertex combinations is a *brute force* algorithm, where all vertex combinations are iterated through. For each iteration, the connectivity of the selected combination is checked, thereby discarding unconnected combinations. The downside of the simple *brute force* algorithm is its low scalability with $O(2^N)$. A more efficient algorithm is *CONSUBG* [21] which exploits the connectivity of the graph, i.e. only connected vertices are traversed after all. As shown in [21], *CONSUBG* is considerably more complex compared to a *brute force* algorithm and therefore is only superior on *sparse graphs*, i.e. graphs where the average number of edges per vertex is much smaller than the total number of vertices.

Finally, each given vertex combination is expanded into a track segment combination in (cf. *getScombs()*). For instance, 2 vertices with each 3 track segments will be expanded to a total of 9 hypotheses. Afterwards, implausible hypotheses are filtered according to the aforementioned spatio-temporal constraints. This is accomplished by iterating over *Scombs* and only adding *plausible* hypotheses to the result set *Scombs'*. A hypothesis is considered *plausible* if it is not overlapping in time Eq. (6) and the estimated velocity between adjacent track segments is realistic (e.g. $< 20\frac{m}{s}$).

C. Correlation metrics

At this point, track hypotheses Θ_j have been generated, which need to be compared to the vehicle sensor data O_i in order to find the best match. Therefore, we define a function $\xi(O_i,\Theta_j)$ which yields a measure for the similarity or *pseudo-distance* of O_i and Θ_j, where smaller values represent a better correlation. Thus, $\xi(O_i,\Theta_j)=0$ indicates a perfect correlation between O_i and Θ_j. In the following, two *pseudo distance* functions $\xi(O_i,\Theta_j)$ are presented.

1) Kalman filter metric: In our previous work [20], we derived correlation metrics based on a standard *Kalman Filter Innovation* N_i. If both odometry measurements (e.g. v, ω) and externally-observed detections (x, y) are fed into the Kalman Filter, then N_i represents the difference between measurement and prediction for each state variable, thus reflecting the correlation between vehicle odometry and external measurements.

The following equation shows pseudo-distance correlation metrics with respect to Euclidean distance (D_1), averaged over a time window of K values:

$$D_1(O_i,\Theta_j)=\frac{1}{K}\sum_{i=1}^{K}\sqrt{N_{x,i}^2+N_{y,i}^2} \tag{9}$$

The metric $D_1(O_i,\Theta_j)$ is applied on every track hypotheses Θ_j in conjunction with the vehicle's odometry sensor data O_i $=\{o_1,..,o_m\}$ and represents a pseudo-distance. Thus, a lower value of D_1 indicates a better correlation between the measured and externally-observed vehicle trajectory.

2) Best fit metric: We introduce a second metric for comparing vehicle odometry data O_i with a track hypothesis Θ_j referred to as *best fit metric* $D_2(O_i,\Theta_j)$.

Therefore, the best fit algorithm computes the minimal distance between a set of absolute position measurements in the track hypothesis Θ_j to a relative trajectory integrated from odometry data O_i. The latter is a relative measurement in the sense that it is given up to an unknown translation and unknown rotation due to unknown initial heading and starting position. As a side effect the best fit yields an estimate of the true initial heading and starting position based on the set of absolute position measurements.

Thus, we have an absolute trajectory $\Theta_j=\{(t_i,x_i,y_i)\}_{i=1}^N$ and a relative trajectory $O_i=\{(\hat{t}_k,\hat{x}_k,\hat{y}_k)\}_{k=1}^M$. The best fit computes the optimal offset vector b and rotation R_α such that the average Euclidean distance $D_2(O_i,\Theta_j)$ between temporally adjacent points of Θ_j and O_i is minimal.

$$D_2(O_i,\Theta_j)=\frac{1}{K}\min_{b,\alpha}\sum_{i=1}^{K}\|(x_i,y_i)-(b+R_\alpha(\hat{x}_i,\hat{y}_i))\|^2 \tag{10}$$

$$R_\alpha(x,y)=\begin{pmatrix}\cos\alpha & \sin\alpha\\ -\sin\alpha & \cos\alpha\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} \tag{11}$$

The minimum can be computed in a two-step approach. Firstly, for given rotation α the offset $b=b(\alpha)$ as a function of the rotation is computed analytically by differentiation of Eq. (10) as

$$b(\alpha)=\frac{1}{K}\left(\sum_{i=1}^{K}(x_i,y_i)-R_\alpha(\hat{x}_i,\hat{y}_i)\right) \tag{12}$$

In the second step, we try out all angles discretized to a required precision (e.g. 0.3 degrees), selecting the angle that minimizes the problem Eq. (10). For approximately rotation symmetric relative trajectory shapes this may result in two optima. Checking that the timestamps at the ends of Θ_j and O_i match allows to select the correct solution.

3) Joint pseudo-distance correlation metrics: Both $D_1(O_i,\Theta_j)$ and $D_2(O_i,\Theta_j)$ represent the average pseudo-distance between the vehicle's sensor data O_i and the track hypotheses Θ_j. However, these metrics do not penalize length differences of O_i and Θ_j. For example, considering $\Theta_1 = \{S_1, S_2, S_3\}$ and $\Theta_2 = \{S_2, S_3\}$, the metrics D_1 and D_2 could favor Θ_2 over Θ_1 although Θ_1 is the correct match. Consequently, we introduce auxiliary metrics for taking the temporal and spatial overlap into account:

$$b_1(O_i, \Theta_j) = \frac{max(\Delta T(O_i), \Delta T(\Theta_j))}{min(\Delta T(O_i), \Delta T(\Theta_j))} \tag{13}$$

$$b_2(O_i, \Theta_j) = \frac{max(\Delta L(O_i), \Delta L(\Theta_j))}{min(\Delta L(O_i), \Delta L(\Theta_j))} \tag{14}$$

as $\Delta T()$ and $\Delta L()$ denote the temporal and spatial size of a trajectory of vehicle O_i and track hypothesis Θ_j, $b_1(O_i, \Theta_j)$ yields the corresponding mismatch ratio of the *lifetime* and $b_2(O_i, \Theta_j)$ of *distance*. Consequently, the ideal value for both metrics is 1.0 in case of identical temporal or spatial length.

Thus, these auxiliary metrics are combined with the pseudo-distance function resulting in the following joint distance functions $\xi(O_i, \Theta_j)$:

$$\xi_1(O_i, \Theta_j) = D \tag{15}$$

$$\xi_2(O_i, \Theta_j) = D^{\gamma_1} \cdot b_1 \cdot b_2 \tag{16}$$

$$\xi_3(O_i, \Theta_j) = D + \gamma_2 \cdot b_1 + \gamma_3 \cdot b_2 \tag{17}$$

ξ_1 only contains the pseudo distance metric without considering the auxiliary metrics b_1 and b_2. In contrast, ξ_2 and ξ_3 incorporate the auxiliary metrics b_1 and b_2 as product (ξ_2) and sum (ξ_3). The factors γ_1, γ_2 and γ_3 can be adjusted to influence the weighting of the individual terms on the joint distance function ξ_2 and ξ_3.

Having derived several joint distance functions $\xi(O_i, \Theta_j)$, they can be used to find the best match out of a set of track hypotheses $\{\Theta_1, \Theta_2, .., \Theta_m\}$ to the vehicle's sensor data O_i by selecting that track hypotheses Θ_j resulting in the minimum distance value.

D. Practical remarks

Some remarks regarding the practical implementation are provided in the following.

1) Calculation of current position: So far, our proposed approach finds the best matching track hypotheses Θ_j for a vehicle that represents the absolute trajectory of the vehicle. In terms of the big picture in the context of a positioning system, we also need to be able to continuously locate and track the vehicle's current position, particularly when moving through *coverage gaps* (cf. Fig. 1 A). To achieve this, all values of Θ_j and O_i can be fed into a Kalman Filter which yields the current position even in coverage gaps and is robust to measurement inaccuracies.

2) Length of time window ΔT_K: The time window ΔT_K influences the amount of buffered data in O_i and S_i. The higher this value, the more data is buffered and used for the generation of track hypotheses Θ_j as well as calculation of the pseudo-distance $\xi(O_i, \Theta_j)$. Increasing ΔT_K leads to the consideration of longer trajectories and potentially more accurate assignments but also significantly increases the computational complexity as there are more possible combinations in the track hypotheses generation. Thus, it is advisable to find a trade-off between assignment accuracy and correctness depending on the specific environment.

3) Identification of multiple vehicles: At the time of this writing, we handle multiple vehicles independently of each other, i.e. each vehicle's sensor data O_i is compared against all hypotheses Θ_j. This is feasible if many unregistered vehicles are driving through the carpark which are not participating in the system, i.e. not providing their sensor data. However, assuming that multiple vehicles participate, the relative trajectory of every vehicle can be utilized for the identification. A straightforward approach is to establish a MxN matrix Ω with M vehicles as rows and N track hypotheses as columns and fill in the values: $\Omega_{i,j} = \xi(O_i, \Theta_j)$. The goal is then to find those assignments $\{i, j\}$ in Ω which result in the minimum sum. This can be done by applying the Hungarian method [18], which however requires a squared matrix, i.e. $M = N$. A workaround would be the padding of non-existing fields with high values so that they are disregarded.

V. Evaluation

In the following evaluation, the performance of the presented tracking-by-identification methodology and its two main components *track hypotheses generation* and *pseudo-distance correlation metrics* are quantitatively determined. The goal is to show how accurate the proposed metrics are able to assign the correct track hypothesis Θ_j to the corresponding vehicle data O_i. Thus, we will first introduce evaluation metrics, explain the test scenario and finally show the performance results of our proposed approach.

A. Evaluation metrics

As mentioned before, our approach is put on top of existing tracking systems (cf. Fig. 2) in order to identify communication endpoints, e.g. registered vehicles which receive their appropriate position information once the global tracking is in progress. Thus, common evaluation metrics for local tracking systems (e.g. *MOT* [22]) are not suitable for our proposed approach. Instead, we propose three different metrics indicating the ratio of correct identifications, the length of incorrect identifications and the ratio of identifier switches.

Given a total time period of correct track hypothesis assignment T_c and one with incorrect assignment T_f, the *Correct Assignment Rate* is defined in % as:

$$CAR = \frac{T_c}{T_c + T_f} \tag{18}$$

Furthermore, the *Maximum Incorrect Assignment Duration* is defined as maximum time interval of incorrectly assigned track hypothesis:

$$\Delta T_{f,max} \tag{19}$$

		Last Track Segment Correct			All Track Segments Correct		
		CAR in %	$\Delta T_{f,max}$ in sec	$IDSWR$ in %	CAR in %	$\Delta T_{f,max}$ in sec	$IDSWR$ in %
ξ_1 (D only)	D_1	80.96	12	11.4	75.44	21	12.1
	D_2	76.62	12	13.28	68.63	20	13.98
	D_1 & D_2	87.9	17	3.64	82.37	17	5.76
ξ_2 (Product with $\gamma_1 = 2$)	D_1	82.84	12	12.81	76.15	21	14.92
	D_2	80.26	10	13.28	70.98	21	14.92
	D_1 & D_2	88.48	16	3.41	82.37	17	6.7
ξ_3 (Sum with $\gamma_2 = \gamma_3 = 0.062$)	D_1	83.31	11	11.16	76.85	21	13.51
	D_2	81.2	15	12.81	72.39	12	14.69
	D_1 & D_2	90.01	17	3.17	83.08	17	6.7

TABLE II
EVALUATION RESULTS OF PSEUDO-DISTANCE CORRELATION METRICS

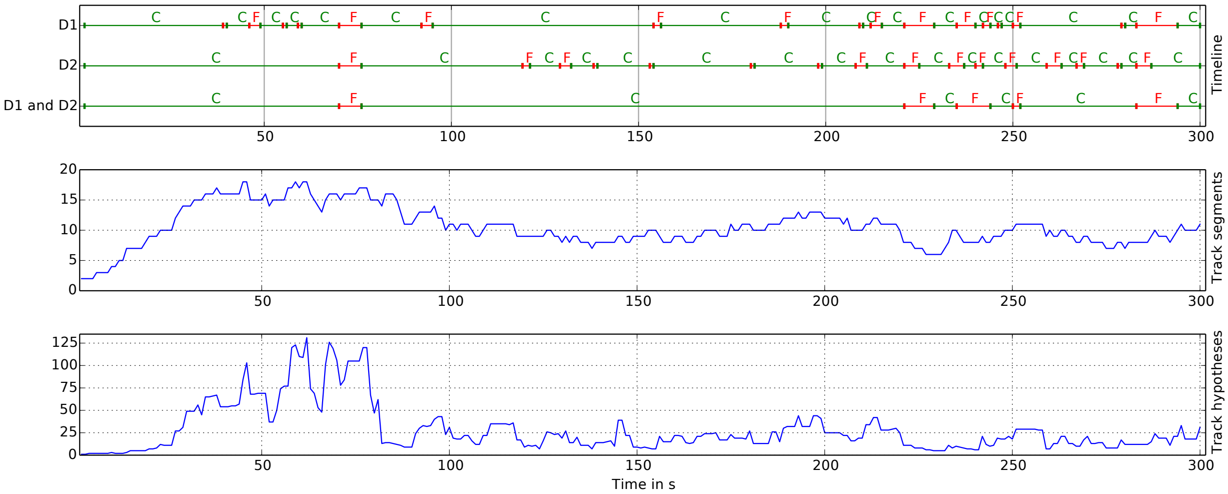

Fig. 3. Evaluation results of pseudo-distance correlation metrics (excerpt, first 300s of total 850s experiment, configuration with ξ_3 and *Last Track Segment Correct*). Top: Identified labels by D_1 (Kalman Filter), D_2 (Best Fit) and D_1&D_2 (combined) approach resp. (correct and false assignments are denoted by green 'C' and red 'F' resp.). Middle: Number of track segments S. Bottom: Number of track hypotheses Θ.

Last but not least, the *ID Switch Ratio* is defined according to the number of ID switches N_{idsw} per total number of identification steps N_t. Therein, $N_t = f_{id} \cdot T_t$, i.e. N_t depends on the frequency of identification f_{id} and the total time period T_t. An ID switch is defined as change of the identified endpoint ID per vehicle. For instance, considering a registered vehicle A which has the correct track hypotheses A assigned when a misidentification to hypotheses B occurs, this is counted as a single identity switch.

$$IDSWR = \frac{N_{idsw}}{N_t} \tag{20}$$

These three metrics are somewhat complementary as different aspects of the tracking-by-identification system are evaluated. First of all, the CAR should be as high as possible, ideally 100% which would indicate a perfect identification, i.e. $T_f = 0$. Secondly, $\Delta T_{f,max}$ shows for a given metric how long the maximum mismatch is and thus indicates the reliability of a metric under challenging conditions. Lastly, $IDSWR$ is a helpful metric for measuring the stability of a given metric. Hence, a high $IDSWR$ indicates that the correct identification is frequently lost and that the determined positions of the endpoints tend to fluctuate resulting in potentially large *jumps*.

B. Test setup and procedures

For the evaluation, we recorded a total of 21 test drives on arbitrary paths through the camera-observed carpark lanes (cf. Fig. 1 A) in our carpark test site (infrastructure described in detail in [10]). Each test drive has a duration between 60s and 140s and contains the vehicle's odometry data and externally observed camera detections. The sampling rate of the vehicle's odometry is $10Hz$. Moreover, the speed of the drives varies between $1.5\frac{m}{s}$ and $5.0\frac{m}{s}$. The drives contain realistic patterns, e.g. the vehicle is in some places slowing down (looking for a free parking spot) but then accelerates to drive to another one.

In order to execute different test scenarios, we have developed a *testing framework* which allows to replay the recorded data into the system at a specified time. Thus, variable combinations of different odometry and camera detections can be replayed. Also, the replayed data is labelled with the actual identifiers (e.g. vehicleA, vehicleB, etc.), which enables us to systematically and reproducibly assess CAR, $T_{f,max}$ and $IDSWR$ of the proposed metrics. Another helpful feature of the *testing framework* is a visualization of the identification results which facilitates the real-time debugging of challenging highly dynamic test cases.

The software is implemented in Java (v1.8.0-05-b13) on a 64 Bit Windows 8.1 OS and runs on a computer with Intel (R) Core (TM) i7-4700MQ (2.4GHz), 16 GB RAM and SSD hard disk.

C. Identification of one out of up to 5 vehicles (1:N)

In the following experiment, between 3 and 5 vehicles are driving through the carpark for a total duration of 14 minutes. According to Fig. 2, a single vehicle is providing its sensor data in order to be identified by the system. Hence, this experiment can be considered as $1 : N$ (registered vehicle : total vehicles) example for global tracking and identification. The tracking-by-identification process is executed every second (i.e. $f_{id} = 1Hz$), which yields a total of 853 assignments. The time window size ΔT_K is set to 30s.

In each identification step, the metrics D_1, D_2 as well as b_1 and b_2 are logged so that different parameters of the joint metrics ξ_1, ξ_2 and ξ_3 can be evaluated on the same *reference data set*. Thus, based on this *reference data set* the optimal weighting of parameters can be obtained that results in the best performance in terms of CAR, $T_{f,max}$ and $IDSWR$.

The test results are presented in Tab. II: Each previously introduced joint distance function ξ_1, ξ_2 and ξ_3 is applied with D_1 (based on Kalman Filter), D_2 (Best Fit) or D_1 & D_2 (combination of Kalman Filter and Best Fit). We evaluate the evaluation metrics CAR, $T_{f,max}$ and $IDSWR$ twice by differentiating which association is counted as *correct*: For *Last Track Segment Correct* it is sufficient if the last (i.e. most recent) track segment is correctly assigned whereas for *All Track Segment Correct* every single track segment in the hypotheses needs to be assigned correctly. The first case is introduced because even if a track hypothesis contains a mismatch but the final track segment is correct, then the vehicle's current position is correctly estimated.

Regarding the combined metric D_1 & D_2, the assignment is only changed if both D_1 and D_2 have identified the same track hypothesis Θ_i (we assume identical last track segment to be sufficient). Otherwise, if D_1 and D_2 point to different track hypotheses, then the previous assignment is retained. In Fig. 3, a timeline of the assignment results for the first 90s of the test is displayed (ξ_3 and *Last Track Segment Correct*). The timeline shows periods of correct and false assignment (marked as *C* and *F* resp.) and illustrates that the D_1 and D_2 in combination are more accurate (we expect higher CAR) as well as more stable (we expect lower $IDSWR$) as D_1 and D_2 need to concur to change the assignment of a track hypothesis. This is confirmed by the results in Tab. II: The combined metric D_1 & D_2 generally outperforms the individual metrics D_1, D_2 in terms of CAR and $IDSWR$. However, this also leads to a longer time interval of incorrect assignment $T_{f,max}$.

Moreover, D_1 (Kalman Filter) performs better than D_2 (Best Fit) in this particular test case. The best joint distance function is the sum ξ_3 which performs better than the product ξ_2 and the individual distance function ξ_1.

Thus, the best result in terms of CAR is 90% for ξ_3 with D_1 & D_2. In this case, $T_{f,max}$ is 17s which is a relatively long continuous time interval for the incorrect assignment. Upon further investigation, we discovered that this is caused by multiple vehicles driving on circular paths at similar speeds as shown in Fig. 4. This leads to the generation of track hypotheses in different views that are very similar and match equally well to the vehicle's relative trajectory.

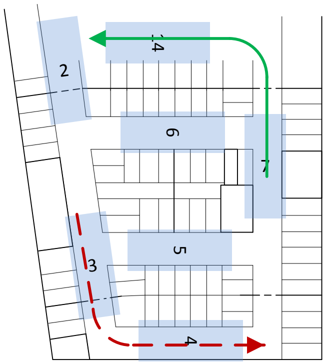

Fig. 4. Example for a difficult scenario where trajectories of two vehicles are very similar (plotted as green straight and red dashed lines resp.).

This example illustrates the limitations of the proposed approach which we expect to have a low performance if all vehicles move in a synchronized manner on identical paths and at similar speeds. This could be the case in crowded environments or traffic jams, where the movements of all vehicles are very similar.

D. Real-time operation

As stated previously, the proposed tracking-by-identification approach is executed once per second in the experiment (i.e. $f_{id} = 1Hz$). However, in many cases a continuous execution is not necessary. Instead, a vehicle can use the system once to obtain an initial starting position which is then updated using odometry and nearby external detections. Nevertheless, our approach can be applied periodically to reinitialize the vehicle's location which helps to correct errors such as ID switches of trackers, drift of odometry measurements, etc.

The average processing time per track hypothesis (and standard deviation) is for D_1 9.8ms (2.3ms) and for D_2 8.4ms (3.1ms). Hence, using both D_1 and D_2 would take 18.2ms (3.86ms) per track hypothesis. In Fig. 3, the maximum number of hypotheses Θ is about 130. Consequently, applying both D_1 and D_2 takes a total of about 2.4s. During the remaining experiment, the number of hypotheses never exceeds 50, so the calculation is completed in less than 1s.

VI. Conclusion and Outlook

We have introduced a tracking-by-identification approach which can be integrated on top of existing tracking systems for globally tracking and identifying vehicles based on relative sensor data. This approach enables to assign a set of external detections to the corresponding vehicle and thus also yields the vehicle's current location without requiring any initialization.

To achieve this, we partitioned the complex data association problem into the two parts track hypotheses generation and correlation metrics. The first algorithm generates track hypotheses of track segments that are compared with the vehicle's relative trajectory by our proposed correlation metrics in the second part, in order to find the best match.

In a detailed evaluation in a realistic testing environment with a challenging data set, we have shown that our proposed approach achieves a correct assignment rate of at least 90% and operates in real-time. Moreover, we investigated test cases of low performance in order to explore the limitations of the approach. For instance, in scenarios where multiple vehicles move in a circular manner at similar speeds, there are long intervals of misassignments of up to 17s. This is due to the fact that multiple generated track hypotheses are of similar shape and fit equally well to the vehicle's relative trajectory.

In conclusion, the proposed approach is ideally suited for environments where vehicle movement is arbitrary and subject to high variations in speed and travelled paths such as carparks. The fact that only relative sensor modalities at the vehicle are required and that surveillance camera infrastructures are often readily available yields a cost-effective positioning solution.

Future research directions include the exploration of additional sensor modalities at the vehicle. Firstly, absolute sensor modalities (e.g. WiFi or Bluetooth positioning) can be employed for situations where the vehicle movements are synchronized. Secondly, as odometry data needs to be accessed at the vehicle's CAN bus, a solution that only works with smartphone sensors would be a promising alternative.

References

[1] M. G. Wing, A. Eklund, and L. D. Kellogg. Consumer-grade global positioning system (gps) accuracy and reliability. *Journal of Forestry*, 108(4):169–173, 2005.

[2] M. Monnerat. Agnss standardization - the path to success in location-based services. *InsideGNSS*, 3(5):22–33, 2008.

[3] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray. Autonomous driving in urban environments: approaches, lessons and challenges. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928), 2010.

[4] R. Mautz. *Indoor Positioning Technologies*, volume 86. Swiss Geodetic Commission, 2012.

[5] M. De Cecco. Sensor fusion of inertial-odometric navigation as a function of the actual manoeuvres of autonomous guided vehicles. *Measurement Science and Technology*, 14(5):643, 2003.

[6] D. Han, S. Jung, M. Lee, and G. Yoon. Building a practical wi-fi-based indoor navigation system. *Pervasive Computing, IEEE*, 13(2):72–79, Apr 2014.

[7] J. Haverinen and A. Kemppainen. Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57(10), 2009. Proceedings of the 5th International Conference on Computational Intelligence, Robotics and Autonomous Systems (5th CIRAS).

[8] J. Borenstein and L. Feng. Gyrodometry: a new method for combining data from gyros and odometry in mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation, 1996*, volume 1, pages 423–428 vol.1, Apr 1996.

[9] Cheng C., Wennan C., A. K. Nasir, and H. Roth. Low cost imu based indoor mobile robot navigation with the assist of odometry and wi-fi using dynamic constraints. In *Proceedings of the IEEE/ION Position Location and Navigation Symposium (PLANS), 2012*, pages 1274–1279, April 2012.

[10] J. Einsiedler, D. Becker, and I. Radusch. External visual positioning system for enclosed carparks. In *Proceedings of the 11th IEEE Workshop on Positioning, Navigation and Communication 2014 (WPNC'14)*, 2014.

[11] A. Ibisch, S. Houben, M. Schlipsing, R. Kesten, P. Reimche, F. Schuller, and H. Altinger. Towards highly automated driving in a parking garage: General object localization and tracking using an environment-embedded camera system. 2014.

[12] A. Ibisch, S. Stumper, H. Altinger, M. Neuhausen, M. Tschentscher, M. Schlipsing, J. Salinen, and A. Knoll. Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded lidar sensors. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2013 (IV'13)*, pages 829–834, June 2013.

[13] D. J. Dailey, F. W. Cathey, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. *Intelligent Transportation Systems, IEEE Transactions on*, 1(2):98–107, Jun 2000.

[14] R. Mautz and S. Tilch. Survey of optical indoor positioning systems. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation 2011 (IPIN'11)*, pages 1–7, 2011.

[15] Xiaogang Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3 – 19, 2013. Extracting Semantics from Multi-Spectrum Video.

[16] Carlo S. Dore, A. Video-radio fusion approach for target tracking in smart spaces. In *Proceedings of the 10th International Conference on Information Fusion*, 2007.

[17] R. Mandeljc, J. Pers, M. Kristan, and S. Kovacic. Fusion of non-visual modalities into the probabilistic occupancy map framework for person localization. In *Proceedings of the 5th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), 2011*, pages 1–6, Aug 2011.

[18] D. Jung, T. Teixeira, and A. Savvides. Towards cooperative localization of wearable sensors using accelerometers and cameras. In *Proceedings of IEEE INFOCOM 2010*, pages 1–9, March 2010.

[19] E. Martin, V. Shia, Posu Yan, P. Kuryloski, E. Seto, V. Ekambaram, and R. Bajcsy. Linking computer vision with off-the-shelf accelerometry through kinetic energy for precise localization. In *Proceedings of 5th IEEE International Conference on Semantic Computing (ICSC), 2011*, Sept 2011.

[20] D. Becker, J. Einsiedler, B. Schäufele, A. Binder, and I. Radusch. Identification of vehicle tracks and association to wireless endpoints by multiple sensor modalities. In *Proceedings of the 4th International Conference on Indoor Positioning and Indoor Navigation 2013 (IPIN'13)*, 2013.

[21] S. Karakashian, B. Y. Choueiry, and S. G. Hartke. An algorithm for generating all connected subgraphs with k vertices of a graph. 2013.

[22] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.