.43944

DMB

DATA MANAGEMENT
AND
BIOMETRICS

NEAR-MISS DETECTION ON TRAFFIC INTERSECTIONS WITH A DISTRIBUTED OVERLAPPING MULTI-CAMERA SYSTEM

Luuk van der Weide

MASTER ASSIGNMENT

Committee:
Dr. Ir. Luuk Spreeuwers (UT)
Dr.Ing. Gwenn Englebienne (UT)
Ir. Michael Dubbeldam (Technolution)

March, 2024

Data Management and Biometrics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Near-Miss Detection on Traffic Intersections with a Distributed Overlapping Multi-Camera System

Luuk van der Weide
University of Twente / Technolution

***Abstract*—Intersections are critical areas where a substantial number of traffic accidents occur. Detecting and analyzing near-miss events is key to signaling a high risk of accidents early and taking appropriate measures. Currently, the acquisition of near-miss event data is limited by time and resources. Edge-based cameras offer a new, powerful method for continuously monitoring these events.**

This study introduces a multi-camera system for near-miss detection. It is designed to operate on a configuration of Technolution's Flowcubes in real-time and on edge. The system leverages rudimentary bounding-box information as input to maintain computational efficiency. It uses a stepwise approach, consisting of middle-point estimation, image-to-world projection, cross-camera association, and trajectory and size estimation. Finally, the resulting shared world-view facilitates the detection and classification of near-miss events.

The system has been tested in both synthetic and real-world scenarios, demonstrating robust performance in deriving a cohesive world model, and promising results for near-miss event detection. It presents a scaleable solution for city-wide traffic safety monitoring with minimal infrastructure investment and reduced reliance on human supervision.

Keywords: Traffic Safety Monitoring, Near-Miss Detection, Tracking-By-Detection, MCMOT, Time To Collision, Post Encroachment Time

I. Introduction

Monitoring road safety is key to being able to decrease accident risks. Especially at intersections, due to the confluence of traffic modalities and travel directions. However, obtaining data for an accurate risk estimation is often limited or resource-intensive.

Earlier safety estimation is usually based on accident count. Therefore, data collection relies on accurate and complete reporting of accidents, and requires multiple accidents to occur before insight into methods of risk mitigation can be achieved.

Many methods aim to use surrogate information, to get a course measure of safety. Examples are the intersection layout, travel directions, and traffic participant counts. Often, this yields a probabilistic risk assessment [1]–[7]. These probabilistic methods have a coarse time resolution of days to months. These analyses lack situation-specific information and therefore are of limited use.

In contrast, near-miss events, which occur when traffic participants come dangerously close to an accident, provide a more accurate indication of safety. As such an event is much closer to an actual accident, it is a far better predictor for safety analysis and less sparse than accidents. However, obtaining this information requires analysis of every traffic participant's trajectory individually. Requiring a different data acquisition method.

Works aiming to accurately track traffic participants have been proposed leveraging aerial, i.e. drone, footage to easily obtain an accurate Birds Eye View (BEV) [8]–[10]. Or BEV is obtained using a fisheye camera positioned above the intersection [11]. These approaches use retrospectively analyzed camera footage, which allows for powerful segmentation and 3d object orientation models to obtain an accurate world model. These methods have proven robust for vehicle positioning. Some of these methods have successfully analyzed safety with time-based metrics [8], [9]. However, both approaches are restrained by the high-effort and drone-battery-constrained data collection, further limiting analysis duration and general scalability to multiple intersections at longer timeframes. Often the hardware costs, required infrastructure, and need for human supervision have posed a continuous safety evaluator infeasible.

Using edge devices, there is a new opportunity to deploy continuous monitoring systems. Technolution's FlowCube is such a device. A FlowCube is a small device equipped with a camera capable of object detection and further processing directly on edge. Multiple devices are combined to obtain increased accuracy and a large coverage area. These devices can be wirelessly connected and mounted on existing infrastructure, significantly reducing both setup and operating costs.

This study introduces a system designed specifically for near-miss event detection at traffic intersections with FlowCubes. Utilizing combined information from these well-positioned cameras, our approach can achieve the needed accuracy online and largely on edge. By focusing on aspects such as traffic participant size estimation in combination with refined trajectory information, this system is engineered to provide accurate safety assessments of intersections by kinematic metrics.

The system leverages the increasing performance of object detection models, such as the model of the YOLO-family [12] in this study. The model facilitates the detection of traffic participants by outputting bounding boxes. Our subsequent system continuously combines a substantial amount of detection data collected across multiple cameras. This information is distilled into the near-miss event count for safety evaluation. To achieve this, a comprehensive method is developed via a modular step-wise approach. A cohesive BEV world model is created by combining the view of multiple cameras. Traffic participant trajectories and sizes are estimated based on the information of multi-angle detections. The calculated kinematic metrics Time To Collision (TTC) and Post Encroachment Time (PET) form the basis of near-miss detection. Finally, this information is combined in the temporal domain to categorize the near-miss events.

This system distinguishes itself by employing a low computational cost approach to multi-camera, multi-object association, which does not depend on the visual resemblance of objects but on spatiotemporal information. The input single-camera tracklets of the system fall under the Tracking-By-Detection paradigm. At high frame-rate and camera overlap this proves to be a powerful approach to single-camera tracking. However, at zones with low overlap or occlusion compromising the detection rate, the system often fails to correctly assign identities. Cross-camera matching steps are structured to be able to filter out such single-camera mistakes, by leveraging the high-confidence information of other view angles. Our work proves by performance comparison the multi-camera system can ”repair” single camera mistakes such as identity switches or identity splits, and improve overall positioning accuracy and robustness.

This research uses a setup of anywhere from 1 up to 6 asynchronous cameras, time synchronization allows for temporal alignment despite varying frame rates (typically between 5 to 10 fps) and frame release times. The system uses known camera parameters, a calibrated extrinsic matrix, and a generic intrinsic matrix without individual camera calibration. The cameras are assumed to have partially overlapping fields of view.

The system is designed to be able to run near real-time, using only limited edge computing processing power. While the aim of our system is to run online at the edge, a delay of 30 seconds is imposed to gain increased accuracy in the tracking and positioning and ultimately near-miss detection. Which is justifiable by the system's monitoring nature, as opposed to a safety-critical system.

Single-camera object detection and tracking is out of the scope of the research, bounding boxes linked via a shared id per camera view are used as the input of this system. All camera views are projected onto a shared worldview. For simplicity in camera-to-world conversion, a flat road surface is assumed. The final goal of the system is to operate in the field continuously, without needed human supervision. The system is designed to be able to trigger a video capture for near-miss events to obtain data for a long-term analysis of the number, severity, and type of incidents in an intersection. To keep the number of fragments to analyze low, an aim of high precision is favorable over higher recall. This paper discusses the system's methodology and its performance evaluation based on tests conducted with both synthetic and real-world data.

In summary, our contributions are:

1) A computationally low-cost method of associating traffic participants cross-camera based on combining single-frame bounding box information.
2) A computationally low-cost method of estimating traffic participant sizes for various modalities solely based on tracking information and multiple projected bounding boxes.
3) A method for automatic detection and rough classification of near-miss events by kinematic metrics based on a unified 2d Birds' Eye View world model.

II. Literature Review

This research addresses multiple sub-problems: tracking, pose and size estimation, and near-miss detection while aiming for edge computing. Often related work overlaps with a part of the method and aim only, seldom with the whole chain and goals. Initially, an overview of mostly similar pipelines is given, in the following sections individual sub-problems are addressed in more detail.

A. Pipelines

In recent years the rise of object detection models has sparked research in object tracking in traffic. Due to the available infrastructure and high safety risk, many works focus on intersection zones. Often the aim of the work is vehicle tracking and city-wide path estimation. Challenges such as the AI City Challenge with a recurrent city-wide tracking assignment [13], [14], have given rise to multiple contributions on the sub-problem of vehicle tracking on intersections.

In approach and setup closest to the tracking method of this research, is a multi-camera tracking-by-detection framework [15], [16], without the near-miss detection step and goal of edge computing. This research also uses a modular approach, with substeps single camera tracking with a YOLO model, and cross-camera association using a world position and Re-ID vector.

A subset of works aim at systems capable of running online or on edge. A system combining radar and a single camera system uses Joint Detection and Tracking, [17], instead of the more popular tracking-by-detection for tracking. The main goal is achieving real-time edge-capable tracking. Compared to DEEPSORT [18] frame rates of up to about 450% higher are achieved, running at 20 fps on a 6-core i5-9600K CPU.

A small selection of work extends the focus of vehicle tracking and localization for safety analysis, [8], [9], [11]. Works closest to this research include a single-camera approach [9], where a 2d BEV bounding box for vehicles is estimated. The BEV rectangular vehicle representation is used for calculating PET, additionally, a conversion from a sporadic conflict count to a statistically induced intersection safety heatmap is proposed. This approach is available at 20 fps using 2 heavy 2080 ti GPUs, mainly needed because of the costly segmentation model, and therefore not suitable for edge computing. An approach using aerial footage of a drone, Automated Roadway Conflict Identification System, ARCIS [8], has been used successfully for near-miss detection using a PET metric. But is very limited in operating time and has no real-time or edge constraints. A well-positioned fisheye camera can also be used instead of multiple cameras [11]. Using a deep learning method, near-miss events are detected coarsely, by distance in image space or sudden deceleration. The system works in real-time, but runs only at 40fps with a powerful Titan V GPU, therefore not being suitable for edge systems.

While several studies have concentrated on vehicle tracking, there remains a gap in implementing an edge-based system precise enough for near-miss detection. Only a handful of studies target near-miss event detection directly, yielding significant insights. Yet, the transition to real-time, edge-capable systems still has to be made.

B. Near Miss Detection Approaches

Although near miss detection is the final step of the system pipeline, understanding how near misses can be detected is vital for the initial steps of the methodology to provide a logical flow for the reader. Near-miss detection aims to detect and classify scenarios in which a collision between two road users could have occurred. Without an actual collision to pinpoint, the detection criteria remain somewhat ambiguous. This ambiguity allows for a variety of detection and estimation methods [19], [20]. Often surrogate measures are used for risk estimation as near-miss detection, these aim to provide risk assessment in addition to data about actual collisions, which is limited.

1) Behavioral Based: Often road users react to an imminent collision, e.g. by changing the speed or performing a lateral movement [21]. If these behaviors are detected, they can be used as a near-miss indicator. The deceleration rate is a used indicator of a near-miss event[11]. Based on the assumption that a large part of vehicle-vehicle accidents occur from a vehicle violating priority rules [22], scenario-based rules can detect a violating vehicle [23], and employ a real-time violation detector, which can be classified as a type of near-miss scenario.

2) Probability Based: Extracted trajectories can be combined with incident and severity rates for vehicle-pedestrian collisions [2], or for vehicle-vehicle collisions [1] to obtain a probability risk assessment. Newer methods use simulation techniques [5], [6], [24], to obtain a probabilistic safety evaluation. Machine learning models [3] or the simpler naive Bayes [4] can also be used to provide statistical risk assessment based on flow statistics. Although probability-based methods are powerful and generalizable and can benefit from accurate real-time tracks, the time frame is always larger than analyzing individual traffic participants.

3) Kinematic Based: Kinematic-based methods use the spatial position and orientation of objects over time to calculate safety metrics. Time To Collision (TTC) is the most common metric for risk assessment [25]–[27], is the time between a predicted collision at current speed and position, also widely used from vehicle perspective automatic braking. In more complex definitions acceleration and steering angle can also be taken into account[27].

Post Encroachment Time (PET), defined as the time difference between two road users overlapping at any point, is another widely used metric. For intersections, where road users regularly cross trajectories, PET is a well-established method for safety evaluation [8]. Similar to the kinematic PET, each cell of an intersection grid can be checked for PET[8], [7]. Typically risk indicator thresholds are in the range of 0 to $2-5$ seconds.[21], [26]. Where below $0.5s$ is classified as severe risk, and above $2.0s$ is low risk. PET and TTC are often used complementary, or even combined into a single metric [28].

Near-miss classification can be roughly compared to accident classification. Often accident analysis focuses on predicting injury. A widely used method uses each vehicle's velocity and the difference in angle to simplify comparison to the single metric Δv [29]. Often general classification relies on scenario description based on trajectories [30], [31]. Both these methods rely on modality and trajectory information.

Our approach uses a kinematic approach, as TTC and PET metrics are best to quantify near-miss events and are a proven established way of predicting collisions and therefore near-miss events. Object properties such as velocity and heading angle required for calculation have the added benefit of being usable for classification and filtering of near-miss events. As accurate positioning and size is vital for the calculation of these metrics, this will be the focus of the following sections.

C. Traffic Participants Tracking

1) Single Camera Tracking: The most popular method in object tracking is the Tracking-By-Detection paradigm [8], [9], [32], [33]. Some research uses a Joint Detection and Tracking method [17], [34] , which can allow a more efficient system by combining detection and tracking networks, but is generally regarded as less modular, more complex, and can decrease performance for complex tracks.

Tracking by detection assumes a known bounding box by a frame-based detection algorithm and focuses on using this bounding box information. A simple approach is proposed in [35], where an intersection over union approach is used for image frame camera tracking. Based on the important assumption of a high frame rate, an intersection over a union tracker proves to be reliable, cheap, and accurate. Further improvements can be achieved by combining with a variable YOLO classification threshold based on position as proposed in [36] at low computational costs.

The starting point of this research is the YOLO detection algorithm, where detections in single-camera domain are linked through an image Kalman filter and IoU combination. This Tracking-By-Detection method is given, and it is not in the scope of the research to be altered.

2) Multi Camera Tracking: Single Camera Tracking (SCT) methods provide some tracking and localization. A multi-camera setup can leverage the benefits of the different camera perspectives. The different angles can greatly increase the localization accuracy. Additionally, errors due to misdetection, image edge bounding boxes, and occlusion can be recognized and filtered out.

Knowing which information or objects from SCT to combine, is nontrivial due to differences in distance, asynchronous cameras, multiple perspectives, and dynamic objects.

a) Matching: Literature often speaks of MCMOT, Multi-Camera Multi-Object Tracking [37]. A common approach is finding an optimal assignment between high-confidence single-camera trajectory pieces, called tracklets. Tracklets consist of multiple detections and span through time, therefore they are often matched over a timeframe. A simpler online approach can be to assign these detections sequentially.

As the system is designed for continuous operation, our approach uses a sequential assignment. Tracklet-based batch processing is possible continuously by dividing in smaller time frames, but adds more complexity and requires more processing power. We still use the available SCT match information in our sequential assignment, but match each measurement instead of each SCT tracklet.

To find matches, a similarity metric is used. Generally, two methods for calculating measurement or tracklet similarity are differentiated between. Spatial-temporal-based matching [33], [38] or additionally matching on appearance information [8], [9], [15], [18], [39].Appearance-based matching aims to recognize learned features and project them into a multi-dimensional space, in which distances between samples can be evaluated. As this is already implemented in SCT and explicitly not a focus of this research, we will focus on Spatial-Temporal based tracking techniques.

Often appearance-based matching is used in addition to spatiotemporal matching, e.g. DEEPSORT [18] and TrackletNet [40]. These methods prove more robust against tracking errors[41]. An alternative to using appearance vectors for occlusions is to use the lighter-weight future trajectory prediction [42]. This prediction can be made by a Kalman filter, or by applying a reinforcement-trained model that assigns B-splines to tracklets as prediction [43].

Some techniques in pedestrian tracking use an occupation map instead of single camera tracklets[44], [45]. By adapting keypoint detection to find the ground point of pedestrians, a BEV centralized occupancy map is generated. This occupancy map is then used to match corresponding pedestrians cross-camera.

The approach of pairwise matching evaluates the similarity between each tracklet or measurement and tries to find a globally optimal match. Similarity metrics include Euclidean distance, angle cosine similarity, or Mahalanobis distance [46]. These similarities are often followed by a matching algorithm, such as the Hungarian method [38].

Often some heuristics are added to improve the performance at various stages in the matching and association.

A partly visible object often leads to reduced accuracy. This can happen to occlusion by other objects, or by the sensor detection boundaries. A common problem arising from vehicles entering and leaving from the camera edge is an identity switch, where a vehicle wrongfully takes over an identity. Turn splitting removes improbable vehicle paths by splitting a tracklet if object orientation suddenly changes significantly [16]. General matching can be boosted, by subdividing lower confidence points of tracklets and reattaching again in the matching step [47]. Other general approaches are to interpolate tracklets for better accuracy before matching [10], or fitting a B-spline as an alternative to interpolation [43]

In single-frame detection, a confidence threshold for bounding boxes is often relatively high to avoid ghost detections, omitting low-confidence detections. However, these low-confidence bounding boxes can be leveraged in combination with a motion predictor, to still assign these bounding boxes in e.g. an occlusion situation [48].

In object direction tracking, systems leveraging situational knowledge often make use of road topology information by creating track completion zones. Tracks should start and end at certain indicated zones or will be removed [16]. While this allows higher performance, at larger intersections often many zone definitions are needed, leading to loss of generalizability. A logical layer of queuing can be added, with the assumption that road users, e.g. for a traffic light, will remain in the same order. Similarly, the number of objects in specific zones can be counted and evaluated. A simple and logical approach for camera clustering is the assumption that any object will not be observed multiple times in one image frame, and set the distance between camera detections to infinite [45].

Our approach does not specify the specific entering and leaving zones, as in [16], to sustain a generally and easily applicable system without complex intersection-dependent annotation. To not interfere with the existing single-camera tracking method, low-confidence bounding boxes are omitted, unlike in [48]. We do use the assumption each object is detected at maximum only once in each frame [45].

D. Pose and Size Estimation

Pose estimation aims to correctly distill vehicle orientation and or size from the available data, which can be leveraged as input in vehicle tracking and be used for more accurate near-miss detection. Frame- and temporal-based methods are differentiated between.

Recent methods use a CNN to estimate the vehicle orientation, often from a single-camera [8], [49]–[51] , or multi-camera [33], [52] A network can be used for keypoint detection and/or segmentation masks. The input for such systems can be a bounding box with multiple extracted keypoints or key regions. The 3D pose is retrieved by the orientation of these keypoints or after combining information from multiple cameras. An often-used combination of networks is Mask-RCNN [53] with occlusion net [49]. Occlusion net aims to localize occluded keypoints. Another method of a rectangular bounding box reconstruction from keypoints to bounding box, can be done based on the assumption that three on-occluded key points can always be detected [8]. Similarly, a segmentation-based 3d shape estimation is carved via a voxel shape-from-silhouette method [33]. By using multi-angles and multi-frames, higher accuracy is achieved.

By leveraging the modality and known size averages of the road user, the orientation and geometrical size constraints

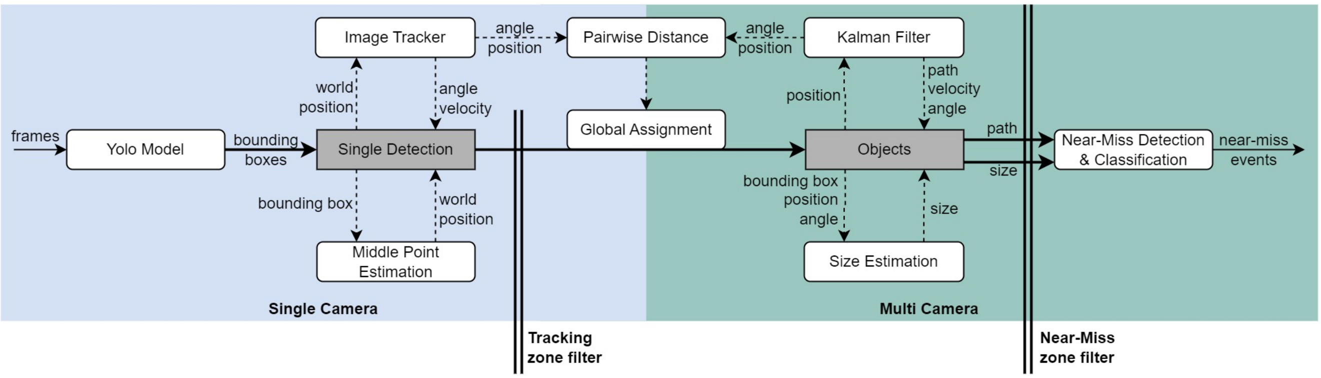


Fig. 1: Method schematic: Detections are initially tracked by each camera, Detections are assigned to objects using a globally optimal assignment across pairwise distances. Objects size and trajectory are updated using these new detections, These are finally used in the near-miss detection and classification step.

within a 2D bounding box can yield a 3D bounding box [51]. Although this popular method uses a CNN for 3d, it could also be implemented in a lower-cost fitting method.

Temporal methods can combine the information of consecutive frames, which is a less computationally costly approach [54], [55]. Common single-frame problems such as an unstable direction through time can be eliminated by filtering [54]. Increased positional and orientation information can be gained by using a fusion of positioning methods [10].

Our approach aims to leverage the computationally cheap spatiotemporal angle determination. Geometrical constraints such as in [51] are then used to determine size. By using the predetermined heading angle, no CNN estimation is needed, reducing complexity. As in [8], we assume three corner points can always be detected. By using detections from multi-cameras and multi-frames accuracy is increased, similarly to [33].

III. Methodology

The methodology is discussed in a step-wise fashion, to reflect the modular approach. An overview of the system is shown in Fig. 1. Initially, setup and configuration is required for the specific intersection, discussed in section ”A Priori Definitions”. The system updates for every single camera frame. Single camera tracking links the detections through time. Via Cross camera association detections of this frame are assigned to the existing objects. Kalman filters combine the detections into a smooth trajectory. Size estimation using all bounding boxes through time of an object completes the world model. This 2D world model is consequently used for near-miss detection using PET and TTC metrics. Finally, the classification step, further filters and labels the near-miss events.

A. A Priori Definitions

The system is developed to be generalizable for various intersections and work in an ad-hoc fashion. However, the system greatly benefits in performance from a number of predefined configurations, which are incorporated into the system. These definitions are therefore chosen to be required as the setup for each intersection.

1) Camera Calibration: The existing system has coarse information (about $\pm 0.5m$ for location,$\pm 0.05rad$ for orientation) on the camera position and orientation. At long distances, especially the height and orientation errors result in significant positional errors in projected world positions. To optimize the system performance, about 10 corresponding points in camera and world domain are used to further calibrate the camera. Because of road marks at the intersection distinct points can be assumed to always be available. A flat road surface is assumed for image-to-world projection. This has a clear limitation at a larger distance from the camera for uneven roads. A projected camera image shows differences with an offset up to $1\ m$, see Fig. 2.

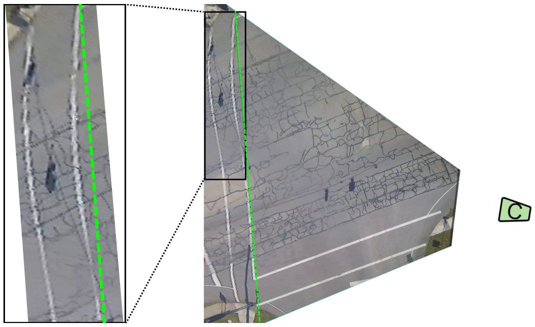


Fig. 2: Influence of the assumption of a flat road surface on positional accuracy, where the green line is the straight line in world domain

2) Zone Definition: In Fig. 3 an example overview of an intersection is shown. The area of interest for near-miss events predominantly lies between pedestrian crosswalks and vehicle stop bars. In principle, the system works for any detection at any range. In practice, accuracy in localization and detection decreases at long distances. Partly this is caused by limited object size in detection of the YOLO model. Additionally, the cameras only have a limited FOV and a limited overlap outside of the intersection area.

The system uses zone definitions to filter out unwanted inaccurate detections and limit the scope of the system to the near-miss area of interest. This is done by the definition of two zones: the near-miss detection is limited to the inner intersection zone. The tracking, where objects can be initialized and tracked is defined to a deeper into the roads opposite to the cameras.

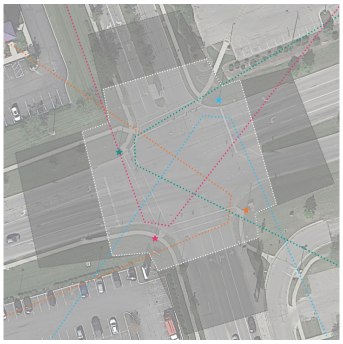


Fig. 3: Example of tracking (grey)- and near-miss (white) zone definition. Colemans satellite view

B. Single Camera Object Detection and Tracking

Although not inherently part of the research scope, single-camera tracking is an important basis of detection. The system uses a custom-trained YOLO model, with 12 classes. Detections are linked inter-frame by IoU (Intersection over Union), which performs well under the frame rate of 5-10 Hz. Additionally, an appearance-based vector is calculated per associated sequence of frames. The appearance-based method is explicitly not used in later stages of cross-camera matching. An image-based Kalman filter is used for linking detections where IoU is not applicable, e.g. due to missed frames.

By applying simple heuristics, detection quality is increased:

- blip filtering, single frame detections are omitted.
- static object filtering, long static detections such as parked cars are assumed to be part of the scenery and omitted.

C. Single Camera Pose Estimation

This research scope starting point is the single camera tracks. To increase quality for matching and later steps the tracks are further processed.

1) Middlepoint Estimation: As a starting point, bounding box information is used to estimate the real-world position of traffic participants. The aim is to find the center of the ground area, Ground Area Middle (GAM), based on the bounding box size and position as illustrated in Fig. 4. The pitch between the camera POV and the bounding box bottom middle is calculated. Theoretically, the GAM can be found at:

$$[BM_x, BM_y + 0.5H_{BB}]\text{for } \theta_{C-BM} = 0 \text{ (straight down)}$$

$$[BM_x, BM_y + 0H_{BB}]\text{for } \theta_{C-BM} = \frac{1}{2}\pi \text{ (at horizon)}$$

By fitting a polynomial to a dataset of known GAMs at a variety of different distances and pitches, a relation between pitch to GAM position is fitted. By using this approximation, the image GAM can be estimated using solely bounding box information. The image GAM can then be projected onto the world plane using the calibrated camera characteristics, as found in section Camera Calibration.

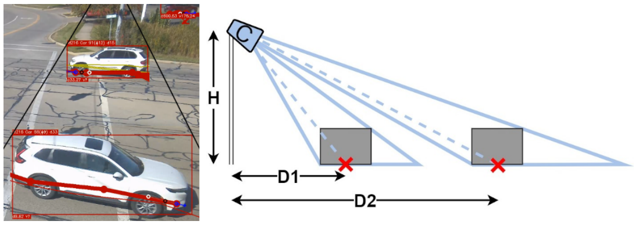


Fig. 4: A schematic of the camera view as an illustration of middlepoint estimation, and directional uncertainties in the longitudinal camera direction. Watkins, t04:52

2) Angle & Velocity Determination: To calculate the detection angle and velocity at any time, the temporal relation between the world GAM is used. To suppress noise in the detection, for each detection, the future and history positions with at least a distance of 1 meter are used to calculate the angle and velocity. The use of a ”future” point can be justified by the total system delay of 30 seconds.

D. Cross Camera Association

The cross-camera association is done sequentially, where detections of a single frame of a single camera are matched in one update. Each object has a corresponding extended Kalman filter with state: $\{x_{world}, y_{world}, v, \alpha_{heading}\}$ In each update, new detections are matched to existing objects. For a traffic participant entering the scene, an unmatched detection initializes a new object.

a) Distance Calculation: The pairwise distance between detection and object is calculated at $t_{detection}$, the object state is extrapolated to this time using the Kalman filter state from $t_{detection-\delta t}$. The distance is largely based on the Mahalanobis distance and the normalized difference in angle, as discussed in section II-C2. In addition, some terms to penalize or incentivize certain matches are added. Parameters are found experimentally.

- The single-camera track ID’s are generally consistent with one ground truth object. To incentivize consistency in assigning multiple detections from the same single camera track to the same object, the following detections with this ID have a lower distance.
- A common mistake is a double detection of one vehicle. The newly initialized object from such a detection

can coincidentally be closer to the following detections. To avoid an identity split, newly initialized objects are penalized in pairwise distance.

- The detection algorithm labels detections by class. Generally, the class assigned is consistent across one ground truth object, and therefore incentivized.

b) Matching: Before the matching step, all detection-object distances larger than a predefined threshold are given a distance of inf. This ensures the Hungarian algorithm only matches objects close to each other, and detection objects with a large distance are not of influence on the global assignment. Generally, each ground truth object is only detected once per frame, allowing at most one detection per object. The Hungarian algorithm finds a global optimum for assigning detections to objects, with the constraint of assigning one detection per object. The assignment benefits from a global optimum under camera bias, caused by e.g. the road height difference discussed in section III-A1, see Fig. 5.

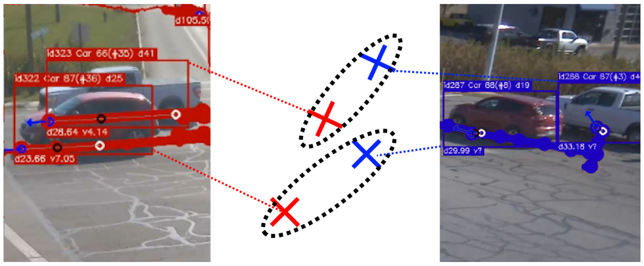


Fig. 5: Example of a situation where matching globally proves beneficial. Watkins [C1 t8:06, C2 t8:10]

Non-matched detections are seen as new objects, and initialize a new object with a new Kalman filter, which is fully instantiated after a threshold of 80 detections. Objects without a match are not automatically discarded. A period of no detections is common, e.g. by occlusion or decreased detection confidence due to partly being out of frame. Therefore a time-out is used for discarding objects, where with each assignment, the timer is reset. The threshold is determined dynamically, based on the number of previous assignments and increases for pedestrian classes to compensate for a lower detection rate.

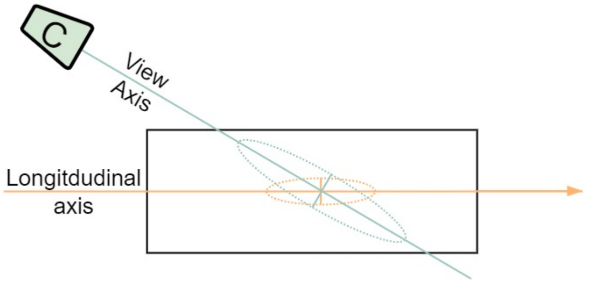


Fig. 6: A schematic of the directionality of the measurement noise based on the (perpendicular) camera view axis, and process noise dependent on the (lateral) longitudinal vehicle direction

c) Object Kalman Filters: With each new assigned detection, the object's Kalman filter is updated. Due to large differences in detection accuracy, a variable measurement covariance matrix is used.

If a vehicle is only partly visible on the camera frame, this is coarsely detected by checking if any of the bounding box corners lie within a threshold on the frame edges. If a bounding box is detected on the edge, often the GAM estimation as discussed in the II-C2, is inaccurate. The measurement covariance is set to infinite, effectively discarding the measurement.

At larger distances, the precision in bounding box coordinates of the detection model decreases. In the view direction, errors in the image Y axis are amplified due to the projection step, as discussed in [56], and further amplified by differences in road height, as discussed in section Camera Calibration. This is illustrated in Fig 4. In the direction lateral to the view direction the variance is mostly based on the detection accuracy. Depending on the distance to the camera and view angle a new measurement covariance is calculated for this update step.

Similarly, for vehicles, the process noise covariance depends on the vehicle's orientation. Lateral motion is less variable than the longitudinal motion. Based on the angle of the current object state, a process noise covariance is calculated. An example of how the covariances apply for a certain detection is shown in Fig 6.

E. Size Estimation

We approach the size estimation as an optimization problem of fitting the ground area of the vehicle inside the world-projected bounding box. Which relies predominantly on geometrical constraints, comparable to the method of [51]. The matching step yields corresponding detections with multiple bounding boxes from various view angles and vehicle poses across time and cameras. Traffic participants are assumed to be shaped as a rectangular beam, of which 3 corners are always assumed to lie on the bounding box [8]. Two examples are depicted in Fig. 7. By minimizing the cost associated with the fit, the size of the vehicle is approximated. This cost is calculated as the weighted sum of distances from each of the vehicle's corner points to the corresponding line of each bounding box.

We denote the ground area middle of the vehicle by $W_{\text{GAM}} = (x, y)$ and α_{heading} as its heading angle. The distance from W_{GAM} to each corner point of the vehicle is represented by R, and the angle offset from the heading to determine the corners is denoted by δ, as in Fig. 8

The corner points of the vehicle, relative to W_{GAM}, R, and δ, are defined as:

$$CornerPoint_i = (x + R\cos(\alpha_{\text{i}}), y + R\sin(\alpha_{\text{i}}))$$

with:

$$\begin{aligned}
\text{Front Right :}\alpha_{\text{FR}} &= \alpha_{\text{heading}} + \delta \\
\text{Front Left :}\alpha_{\text{FL}} &= \alpha_{\text{heading}} - \delta \\
\text{Rear Right :}\alpha_{\text{RR}} &= \alpha_{\text{heading}} + \delta + \pi \\
\text{Rear Left :}\alpha_{\text{RL}} &= \alpha_{\text{heading}} - \delta + \pi .
\end{aligned}$$

Each bounding box is defined by four lines, each with its own weight. The weight vector for the lines of a bounding box is denoted by $\mathbf{w} = \{w_T, w_R, w_B, w_L\}$, where each w corresponds to the weight of a bounding box line based on the image domain Top, Right, Bottom and Left lines. Since the top line is highly influenced by the vehicle height, it is omitted by using a weight of 0, as the bottom line is generally more accurate, a higher weight is given.

Depending on vehicle orientation: α_{heading} and the view-angle of the camera, each corner point is assigned to a corresponding line. This assignment is done based on the difference of α_{heading} and $\alpha_{\text{BB-line}}$. Where depending on the quadrant in which this difference lands, lines are assigned to the corners.

The weighted distance D_i from a corner point $C = (c_x, c_y)$ to the closest point on a line of the bounding box can be calculated using equation 1.

$$D_i = w_i \cdot \frac{|a \cdot c_x + b \cdot c_y + c|}{\sqrt{a^2 + b^2}}, \tag{1}$$

where $a = \sin(\theta_i)$, $b = -\cos(\theta_i)$, and c is derived from the line's equation in world view.

Given multiple bounding boxes, each with its own set of weights, the total cost function $T(R, \delta)$ for fitting a vehicle within these bounding boxes is the average of the weighted costs for each bounding box. For a total of N bounding boxes, each with a weight W_j for the j-th bounding box, the total cost is given by equation 2.

$$T(R, \delta) = \frac{1}{N} \sum_{j=1}^{N} W_j \sum_{i=1}^{4} D_{i,j}(R, \delta), \tag{2}$$

where $D_{i,j}(R, \delta)$ is the weighted distance from the corner points of the vehicle to the lines of the j-th bounding box.

The optimization process involves adjusting the parameters R and δ to minimize the averaged total cost $T(R, \delta)$.

F. Near-Miss Detection

For the final near-miss detection we use both Post Encroachment Time (PET) and Time To Collision (TTC). To reduce complexity and reduce false positives vehicles are extrapolated over their trajectory. The future trajectory is found by delaying the near miss detection by e.g. 15 seconds, which is possible due to the near-realtime flexibility, discussed in section Introduction. A projected collision is identified by overlapping ground area. PET times are found by comparing object "1" at detection time T_d, to all history detections of other objects to a maximum time-window, see Fig. 10. The minimum $|\Delta t|$ between objects is stored as PET. For the Time To Collision, a window of interest is determined as $\frac{\Delta L_{path}}{v_d} \leq \Delta t_{max}$. All object areas with a $t_1 = t_2$ are compared. The minimum Δt between objects is stored as TTC.

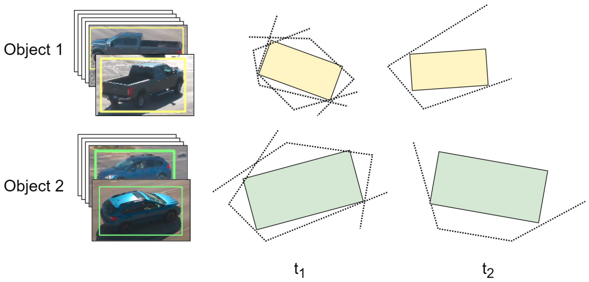


Fig. 7: An example of a collection of detections of one object, bounding boxes across time and cameras are projected to the world domain

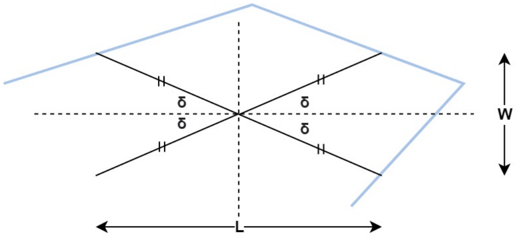


Fig. 8: An example of the optimization within one bounding box, with 4 vehicle corners. The angle and length are optimized. Note that the bounding box top line is not guaranteed to lie on a vehicle corner, and is therefore given weight 0 and omitted in the drawing

G. Near-Miss Classification

TTC and PET are limited descriptors for a variety of near-miss events. To further classify and filter to near-miss events of interest, general classifications are used.

Simple classification is done based on modalities, such as vehicle-vehicle or vehicle-pedestrian accidents.

Based on the proposed Δv for the severity of accidents [29], we use a similar metric where the unknown mass of traffic participants is omitted in equation (3):

$$\Delta V = \frac{1}{2}\sqrt{V_1^2 + V_2^2 - 2V_1V_2\cos(\theta_{\Delta 1-2})} \tag{3}$$

To account for the time difference to the projected collision, we decrease each traffic participant's speed by 1m/s^2 for the time difference to calculate the projected ΔV. Finally, to obtain the "severity score", we increment ΔV by 20 for cars colliding with vulnerable road users (VRU's) such as pedestrians and cyclists. Over time, we group multiple triggers that occur between two road users, which allows for further analysis.

Lastly, a rough scenario-based classification is used [30], [31]. Based on the initial and final orientation of a vehicle, a rough classification to straight, left-turn or right-turn can

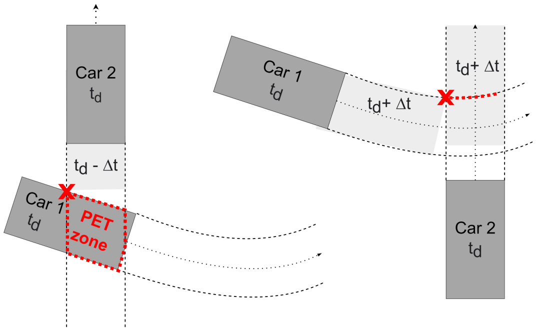


Fig. 10: A schematic of the used metrics, PET left, and TTC right.The red cross indicates minimum Δt for the frame. Note: The schematic at Δt indicates the shortest PET time, the PET is not necessarily constant in the PET zone.
Note: The schematic at Δt indicates the TTC with minimum δt of all extrapolated collision points

be made. Combining these directions and the difference in orientation during collision, the predicted type of accident is categorized.

IV. Experiments

In correspondence with the contributions of this work, the positioning, the cross-camera association, and the size estimation are evaluated. While the collection of severe near-miss events is limited, it suffices for validation of the system. Finally, the operational efficiency is assessed to confirm the system's real-time capabilities on edge devices. First, the datasets and configuration are elaborated on.

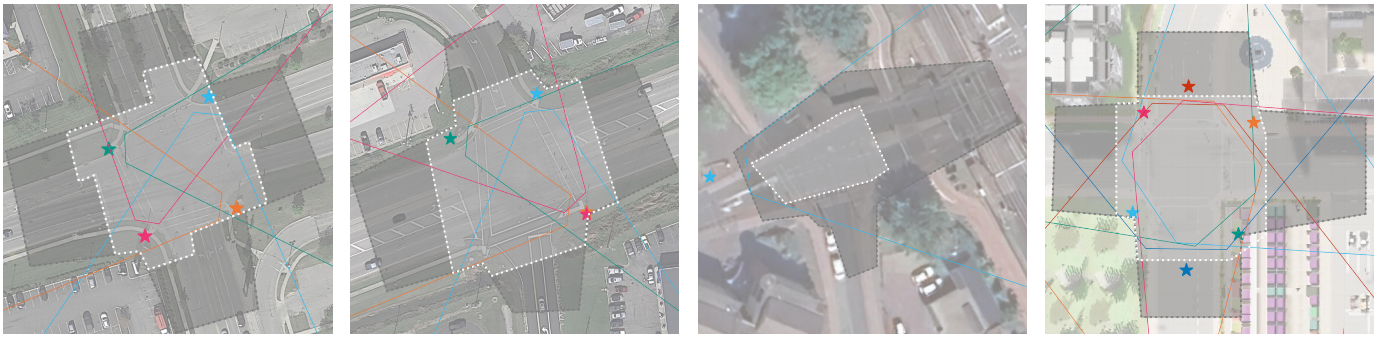
A. Experimental setup

1) Datasets: In the evaluation of the system, multiple datasets are used. An overview of the intersections is given in Fig. 9.

a) Synthehicle: The Synthehicle [57] dataset is a synthetic dataset made in the CARLA engine. The dataset simulates intersections with up to six cameras. We use various configurations: 2 cameras on opposite corners, the 4 corner cameras, or all 6 cameras. Object world position and bounding box are automatically computed based on the 3d models of traffic participants. This allows a large and accurate set of ground truth annotations for performance evaluation. The dataset has 63206 annotated detections of various traffic participants. Assuming there are no errors in the labeling, we use this dataset as a best-case scenario dataset as a theoretical top performance benchmark.

b) FlowCube Data: The system's performance is evaluated further on the custom FlowCube dataset. This dataset is created based on video data from FlowCubes and is comparable to the real-world circumstances the system is designed for. As the object detection is identical to the real-time system under normal operation, the in-the-field system performance is expected to match the performance on this test set. Two intersections are used, Colemans and Watkins. At 30-second intervals, all completely in-screen vehicles are annotated. Annotation is done by assigning a 4 sided polygon, with a vertex on each vehicle corner. As shown in Fig. 11a. To avoid overestimating positioning performance, stationary vehicles are annotated only once. The ground truth width and length, and ground area middle are calculated from this polygon. 104 vehicles are annotated. To assess the tracking performance, detections across cameras are linked to the same traffic participant. In total 3330 detections are annotated. Finally, the dataset is manually checked for severe near-miss events. A near-miss event is classified manually, based on the visible distance and speed between the vehicles and the reaction of traffic participants [21]. However, no severe near-misses are found in the dataset.

c) Single Camera Data: The small amount of FlowCube multi-camera data limits the amount and severity of near-miss events that can be found. To further evaluate near-miss events, additional video data acquired by GoPro cameras is used. 20 hours of data is manually checked for near-miss events. We use a 36-minute fragment from the dataset at the intersection "Hilversum", as this contains the three most severe near-miss



(a) FlowCube dataset: Colemans, 4 min, 40°14'15.7"N, 83°20'58.9"W

(b) FlowCube dataset: Watkins, 8 min, 40°14'23.6"N, 83°20'30.5"W

(c) GoPro dataset: Hilversum, 36 min, 52°13'44.4"N, 5°10'44.4"E

(d) Synthehicle Dataset: Town05, 3 min, [57]

Fig. 9: Overview of the intersections of the various datasets. A grey tracking zone and white near-miss zone are indicated. Cameras and their corresponding field of view are depicted as stars and lines.

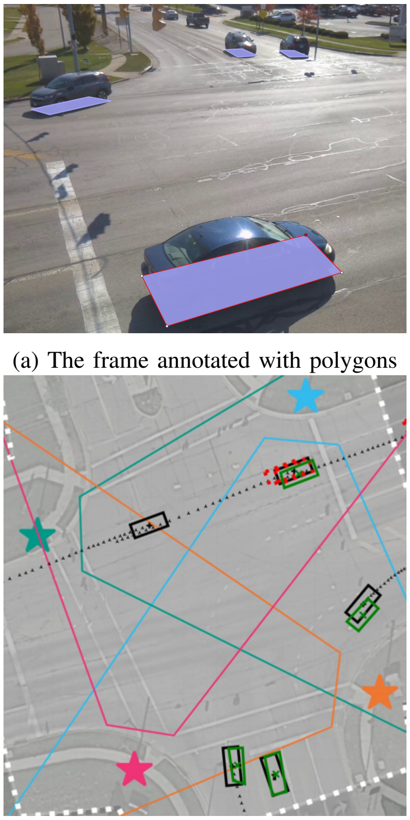
Dataset	Cameras	Position Precision	Association Performance	Size Precision		
		$\Delta P[m] \downarrow$	$AssA \uparrow$	$\Delta L \downarrow$	$\Delta W \downarrow$	$IoU \uparrow$
Synthehicle	2	0.66	0.991	11.6%	6.8%	0.57
Synthehicle	4	0.44	0.991	10.8%	5.9%	0.69
Synthehicle	6	0.36	0.992	10.6%	5.1%	0.74
FlowCube	4	0.72	0.966	9.5%	10.6%	0.61

TABLE I: Results table: The system's performance tested across various datasets and cameras. The average error in precision is always given with respect to the world ground truth. The error in length and width is calculated relative to the ground truth.

Method	Dataset	Cameras	Position Precision	Size Precision
			$\Delta P[m] \downarrow$	$IoU \uparrow$
Baseline	FlowCube	1	2.09	x
Ours	FlowCube	4	0.72	0.61
CAROM [33]	custom	4	0.79	x
Abdel-Aty [8]	location 1	1	0.48	0.55
Abdel-Aty [8]	location 2	1	0.68	0.28

[x] Not available

TABLE II: Comparison table: Comparison is done between several SOTA methods. Note that the datasets are different.



(a) The frame annotated with polygons

(b) The BEV view, ground truth (green) and system output (black)

Fig. 11: A image annotation and its projected BEV view. The annotation is done on the top right (cyan) camera, indicated by a star and field of view, Colemans dataset [t03:01]

events. Although the available material is single camera, it can still be used as input for our system.

2) Setup: As the system is designed to be continuously effective, a warm-up of at least 60 seconds is used before measurements start. This ensures sufficient history data for angle calculation in single camera domain, and detections for size estimation. The cameras are synchronized, for each simulation, all selected cameras are active for the entire time fragment.

B. System Performance

1) World Position Precision: As a single reference point for positioning, the center of the vehicle or "Ground Area Middle" (GAM) is used. The precision is always evaluated in the world domain. The evaluation is done for all points inside the near-miss zone, as discussed in sec. Zone Definition.

An example of the output of a single frame is shown in Fig. 11b. The average Euclidean distance to the ground truth is given in table I. Results show an increase in accuracy with the increase in the number of cameras, indicating the system benefits from multiple cameras.

A comparison to the baseline and methods with comparable datasets is given in table II. The performance of the baseline method is given, which simply projects the bottom middle point of the bounding box to the world domain. Compared to the baseline, logically, a large increase in accuracy is gained. The system shows comparable performance to a monocular segmentation-based system [8], and has a better performance than the CAROM framework [33].

2) Cross Camera Association Performance: The cross-camera association is calculated by using a subpart of the popular Higher Order Tracking Accuracy (HOTA) metric [58], the Association Accuracy (AssA), which is a metric aimed at evaluating the association performance without the influence of detection performance. As the detection step is not part of this research, a metric aimed solely at association performance is favorable.

Results are given in table I. The association is largely correct. Again, the system shows an improvement in association performance when using more cameras. At 6 cameras the number of association errors is halved with respect to the single camera.

The system's enhanced robustness is demonstrated by its performance under unexpected conditions. The system should ignore irrelevant objects. This is achieved by setting a threshold for the number of detections of objects; any object that doesn't meet this criterion, due to inconsistent detection across frames and cameras, is not instantiated. As discussed in sec. III-D0c.

For instance, in the Colemans dataset, there is an example case involving a vehicle carrier truck, illustrated in Fig. 12.The YOLO model is not trained to recognize this type of vehicle. The resulting sporadic detections of vehicles on the carrier, result in the association algorithm failing to consistently track the objects across different frames and cameras, and excluding the detections. Therefore, none of the vehicle carrier objects negatively impact the system performance.

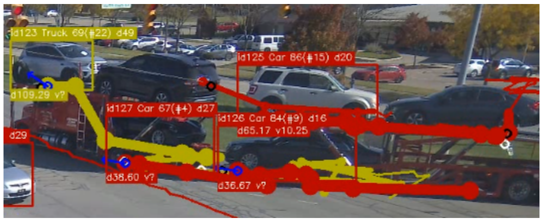


Fig. 12: A vehicle carrier as an example of an anomaly, the carried vehicles are detected at inaccurate positions

3) Size Precision: The estimated sizes are evaluated via three metrics. The IoU with the ground truth ground area, is a straightforward measure used often [8], [9]. To further decompose the size, and omit inaccuracy in position also the relative error in both the width and length estimation is evaluated. An example of the output is shown in Fig. 11b. Results are given in table I.

The experiments show the performance is high for the ideal synthetic data, reaching highest IoU with 6 cameras. The FlowCube data performs slightly lower. The system shows to be robust across different datasets. The scores are compared to other methods in table II. Our method scores than the computationally costly single frame keypoint estimation model of [8].

4) Near-Miss Detection Performance: Because of the scarcity of near-miss events in the limited multi-camera FlowCube data, we make the assumption that a low PET or TTC value, if calculated correctly, will indicate a near-miss event. This way, we can estimate the quantitative performance of the system by analyzing the correctness of the PET and TTC metrics. By analyzing system PET and TTC output between two objects, a true case indicates an actual area of overlap in future or history, a postive case indicates that the system accurately detected such a case. The results of this surrogate performance for a maximum time-difference of $2.5s$ can be found in table III.

5) Near-Miss Detection Validation: To qualitatively validate the near-miss detection, we compare the system's output with manually labeled near-miss events.

Method	PET/TTC triggers	Trigger Performance	Runtime
	#	$precision \uparrow$	$[s]$
FlowCube Multi-Camera	101	0.87	240

TABLE III: PET and TTC trigger performance, all trigger groups with $\Delta T < 2.5s$ are checked on actual overlap

For this analysis, near-miss trigger groups are filtered to remove false positives and low-severity near-misses. With the minimum δt so that: $0.01 > \delta t > 1.5$, severity score> 7.5 and at least 10 triggers per group.

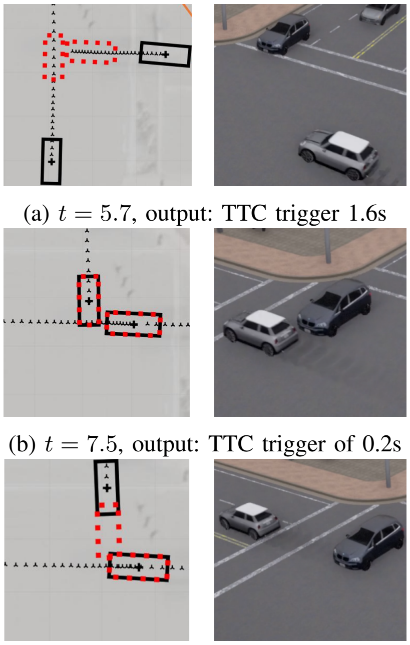
- FlowCube Dataset: No severe near-miss events were observed. Within a span of 12 minutes (4 Colemans, 8 Watkins), the system identified two near-miss events, one low-severity near-miss and one false positive.
- Synthehicle Dataset: A single severe near-miss event was observed and correctly detected by the system over a 3-minute period, as illustrated Fig. 13).
- Single Camera Hilversum Dataset: Three severe near-miss events were observed. The system detected 14 near-miss events across 36 minutes. The three observed severe events were ranked highest based on severity scores and are sequentially presented in Fig. 14. All remaining triggers were true positives of low severity.

The system shows the near-miss events are correctly detected. The calculated severity score helps in the filtering of the near-miss events. The system rarely wrongly detects a near-miss event, however, differentiating between low-severity near-misses remains a challenge.

6) Operational Efficiency: To evaluate the real-time edge performance of the system, a test sequence of 100 seconds, comprising of 1000 frames per each of the cameras was conducted. This sequence is from the colemans dataset, and consists of 4 cameras. Throughout the testing period, the environment consistently contained at least 15 objects, ensuring a substantial processing load.

As the system is currently in its prototype phase, it utilizes relatively slow pandas dataframes for data loading, storage and evaluation. This design decision has significantly affected the performance. To precisely evaluate the real-time capabilities of the system and identify performance bottlenecks, the Python cProfile module was used for an analysis of the subsystems.

During the performance evaluation, on a single thread of an i5-1245U processor, the system registered a total computation time of 201.72 seconds, or 5 iterations per second. A performance breakdown is presented in table IV that specifically omits the time-consuming dataframe operations. This exclusion is intentional, aiming to highlight the processing efficiency of the system's core components. It should be noted that replacing pandas dataframe methods will reduce but not eliminate processing overhead. According to our assessments, with adaptations, the chosen methods have the potential to operate on an edge computing system.



(a) $t = 5.7$, output: TTC trigger 1.6s

(b) $t = 7.5$, output: TTC trigger of 0.2s

(c) $t = 8.0$, output: PET trigger of 0.4s

Fig. 13: The system detecting a near-miss event in the synthetic data. Black boxes are traffic participants with a black dotted trajectory, the red dotted boxes is the projected moment of collision, synthehicle dataset

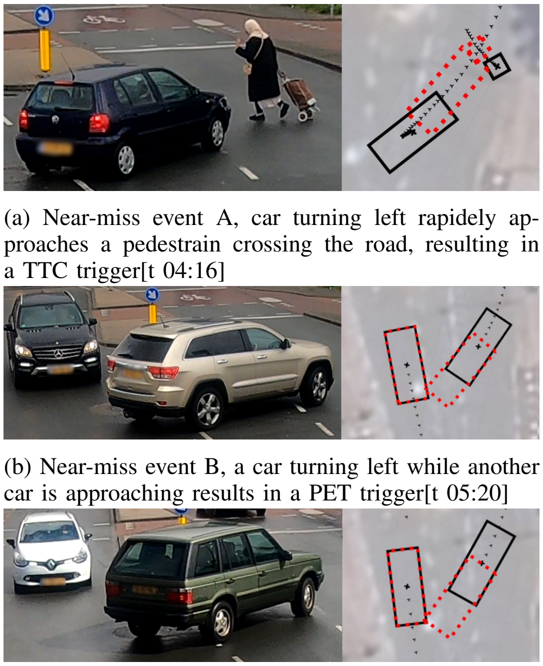
Function	calls [#]	Total Time [s]	Time Per Call [s]
Bounding box projection	4000	0.28	7.07E-05
Image to world	4000	0.37	9.24E-05
Matching: cost matrix	3934	10.97	2.79E-03
Matching: assigment	3934	0.03	6.92E-06
Kalman predict	183829	2.75	1.50E-05
Kalman update	91914	10.88	1.18E-04
Size Estimation	2948	6.68	2.27E-03
Near-miss detection	996	22.37	3.55E-04

TABLE IV: System Profiling: Processing times of various methods of a 100-second time-frame of 4 cameras of the ”Colemans” dataset. Note that time-consuming dataframe operations are excluded.

V. Conclusions and Limitations

We developed an automated monitoring system for tracking traffic participants, estimating their sizes, and detecting near-miss events at intersections using a multi-camera setup with overlapping views. The system operates effectively across various camera configurations, independent of their timing, enhancing robustness and accuracy. Our approach synthesizes multiple camera perspectives into a unified Bird’s Eye View (BEV) world model.

A study with synthetic and in-the-field data across multiple intersections has validated that the system can match and track multiple modalities cross-camera. The method effectively leverages rudimentary bounding box information to form a cohesive and robust worldview. With this, a solid basis for accurate kinematic calculations is formed. Compared to the baseline method, a factor 2.90 increase in positional accuracy is achieved. Size and position estimation, with an average IoU of 0.61, proved to be more accurate than some single-frame methods utilizing computation-intensive segmentation models. As a consequence of the accurate positioning, the PET and TTC can be seen as accurate predictors for collision risk, and can be used in further research for analysis based on impact velocity, angle, and contact point.



(a) Near-miss event A, car turning left rapidely approaches a pedestrain crossing the road, resulting in a TTC trigger[t 04:16]

(b) Near-miss event B, a car turning left while another car is approaching results in a PET trigger[t 05:20]

(c) Near-miss event C, a car turning left while another car is approaching results in PET trigger [t34:35]

Fig. 14: Three examples of near-miss events detected by the single camera system, Black boxes are traffic participants with a black dotted trajectory, the red dotted boxes traffic participants positions at the projected collision, Hilversum dataset

We validated the system’s near-miss detection capability by manually analyzing the system output, confirming its reliability. Using severity metrics, classification, and video analysis, it has the potential to offer detailed insights into safety risks. Field deployment of this system could help to pinpoint the most prevalent hazards, contribute to accident prevention, and reduce the number of incidents.

Future work includes further increasing robustness for multiple modalities such as trucks, by tuning both the detection model and method. The research could benefit from performing a field study for several weeks. Additional near-miss data can be acquired after setting up a video trigger and storing method. Manual classification of such fragments allows a performance assessment of the system. The fragments would comprise a larger dataset that would allow for improved classification and filtering by the severity of near-miss events.

References

[1] A. S. Hakkert and D. Mahalel, "Estimating the number of accidents at intersections from a knowledge of the traffic flows on the approaches," *Accident Analysis & Prevention*, vol. 10, no. 1, pp. 69–79, 1978, ISSN: 0001-4575. DOI: https://doi.org/10.1016/0001-4575(78)90009-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/000145757890009X.

[2] K. T. Yasuhiro Matsui Masahito Hitosugi and T. Doi, "Situations of car-to-pedestrian contact," *Traffic Injury Prevention*, vol. 14, no. 1, pp. 73–77, 2013, Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15389588.2012.678511. DOI: 10.1080/15389588.2012.678511. [Online]. Available: https://doi.org/10.1080/15389588.2012.678511.

[3] J. Hu, M.-C. Huang, and X. Yu, "Efficient mapping of crash risk at intersections with connected vehicle data and deep learning models," *Accident Analysis & Prevention*, vol. 144, p. 105 665, 2020, ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2020.105665. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457519319062.

[4] A. Theofilatos, C. Chen, and C. Antoniou, "Comparing machine learning and deep learning methods for real-time crash prediction," *Transportation Research Record*, vol. 2673, no. 8, pp. 169–178, 2019, _eprint: https://doi.org/10.1177/0361198119841571. DOI: 10.1177/0361198119841571. [Online]. Available: https://doi.org/10.1177/0361198119841571.

[5] Y. Ali, R. D. Chauhan, S. S. Arkatkar, and A. Dhamaniya, "Application of empirical & simulated vehicle trajectories in risk assessment at signalized intersection," *Transportation Research Procedia*, vol. 62, pp. 782–789, 2022, ISSN: 2352-1465. DOI: https://doi.org/10.1016/j.trpro.2022.02.097. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146522002241.

[6] X. Shen and P. Raksincharoensak, "Statistical models of near-accident event and pedestrian behavior at non-signalized intersections.," *Journal of applied statistics*, vol. 49, no. 15, pp. 4028–4048, 2022, Place: England, ISSN: 0266-4763 1360-0532. DOI: 10.1080/02664763.2021.1962263.

[7] Y. Ma, X. Qin, O. Grembek, and Z. Chen, "Developing a safety heatmap of uncontrolled intersections using both conflict probability and severity," *Accident Analysis & Prevention*, vol. 113, pp. 303–316, 2018, ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2018.01.038. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457518300447.

[8] M. Abdel-Aty, Y. Wu, O. Zheng, and J. Yuan, "Using closed-circuit television cameras to analyze traffic safety at intersections based on vehicle key points detection," *Accident Analysis & Prevention*, vol. 176, p. 106 794, 2022, ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2022.106794. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457522002299.

[9] Y. Wu, M. Abdel-Aty, O. Zheng, Q. Cai, and S. Zhang, "Automated safety diagnosis based on unmanned aerial vehicle video and deep learning algorithm," *Transportation Research Record*, vol. 2674, no. 8, pp. 350–359, 2020, 350, ISSN: 0361-1981. DOI: 10.1177/0361198120925808.

[10] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, *GIAOTracker: A comprehensive framework for MCMOT with global information and optimizing strategies in VisDrone 2021*, _eprint: 2202.11983, 2022.

[11] X. Huang, T. Banerjee, K. Chen, N. Varanasi, A. Rangarajan, and S. Ranka, "Machine learning based video processing for real-time near-miss detection," in *Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS,*, Backup Publisher: INSTICC, SciTePress, 2020, pp. 169–179, ISBN: 978-989-758-419-0. DOI: 10.5220/0009345401690179.

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020. arXiv: 2004.10934. [Online]. Available: https://arxiv.org/abs/2004.10934.

[13] M. Naphade, S. Wang, D. C. Anastasiu, *et al.*, "The 6th AI city challenge," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE Computer Society, Jun. 2022, pp. 3346–3355. DOI: 10.1109/CVPRW56347.2022.00378.

[14] M. Naphade, S. Wang, D. C. Anastasiu, *et al.*, "The 5th AI city challenge," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2021.

[15] A. Specker, D. Stadler, L. Florin, and J. Beyerer, "An occlusion-aware multi-target multi-camera tracking system," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 4168–4177. DOI: 10.1109/CVPRW53098.2021.00471.

[16] A. Specker, L. Florin, M. Cormier, and J. Beyerer, "Improving multi-target multi-camera tracking by track refinement and completion," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 3198–3208. DOI: 10.1109/CVPRW56347.2022.00361.

[17] P. Emami, L. Elefteriadou, and S. Ranka, "Long-range multi-object tracking at traffic intersections on low-power devices," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2482–2493, 2022. DOI: 10.1109/TITS.2021.3115513.

[18] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *CoRR*, vol. abs/1703.07402, 2017. arXiv: 1703.07402. [Online]. Available: http://arxiv.org/abs/1703.07402.

[19] S. M. S. Mahmud, L. Ferreira, M. S. Hoque, and A. Tavassoli, "Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs," *IATSS Research*, vol. 41, no. 4, pp. 153–163, 2017, ISSN: 0386-1112. DOI: https://doi.org/10.1016/j.iatssr.2017.02.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0386111217300286.

[20] L. N. Peesapati, M. P. Hunter, and M. O. Rodgers, "Can post encroachment time substitute intersection characteristics in crash prediction models?" *Journal of Safety Research*, vol. 66, pp. 205–211, 2018, ISSN: 0022-4375. DOI: https://doi.org/10.1016/j.jsr.2018.05.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022437517302281.

[21] M. H. Martens and R. Brouwer, *LINKING BEHAVIORAL INDICATORS TO SAFETY: 2 WHAT IS SAFE AND WHAT IS NOT?* 2011.

[22] A. Martinez, H. Evdorides, C. Naing, *et al.*, "Accident causation and pre-accidental driving situations. part 2. in-depth accident causation analysis," Jan. 2008.

[23] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Evaluating risk at road intersections by detecting conflicting intentions," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4841–4846. DOI: 10.1109/IROS.2012.6385491.

[24] J. Bao, P. Liu, and S. V. Ukkusuri, "A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data," *Accident Analysis & Prevention*, vol. 122, pp. 239–254, 2019, ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2018.10.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457518303877.

[25] R. Ke, Z. Cui, Y. Chen, M. Zhu, H. Yang, and Y. Wang, *Edge computing for real-time near-crash detection for smart transportation applications*, _eprint: 2008.00549, 2021.

[26] H. Kataoka, T. Suzuki, S. Oikawa, Y. Matsui, and Y. Satoh, *Drive video analysis for the detection of traffic near-miss incidents*, _eprint: 1804.02555, 2018.

[27] L. Zhang, J. Yuan, K. Yuan, J. Hong, H. Ding, and H. Chen, "Automated braking decision and control for pedestrian collision avoidance based on risk assessment," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 3, 2022, ISSN: 1939-1390. DOI: 10.1109/MITS.2021.3098618.

[28] N. Nadimi, H. Behbahani, and H. Shahbazi, "Calibration and validation of a new time-based surrogate safety measure using fuzzy inference system," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 3, no. 1, pp. 51–58, 2016, ISSN: 2095-7564. DOI: https://doi.org/10.1016/j.jtte.2015.09.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095756415200153.

[29] R. Tolouei, M. Maher, and H. Titheridge, "Vehicle mass and injury risk in two-car crashes: A novel methodology," *Accident; analysis and prevention*, vol. 50, May 2012. DOI: 10.1016/j.aap.2012.04.005.

[30] A. N. S. Institute. "Manual on classification of motor vehicle traffic accidents." (2007), [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/07D16 (visited on 02/25/2024).

[31] N. T. Agency. "Guide for the interpretation of coded crash reports from the crash analysis system (CAS)." (2016), [Online]. Available:

https://www.nzta.govt.nz/assets/resources/guide-to-coded-crash-reports/docs/guide-to-coded-crash-reports.pdf (visited on 02/25/2024).

[32] L. Wen, Z. Lei, M.-C. Chang, H. Qi, and S. Lyu, "Multi-camera multi-target tracking with space-time-view hyper-graph," *International Journal of Computer Vision*, vol. 122, no. 2, pp. 313–333, Apr. 1, 2017, ISSN: 1573-1405. DOI: 10.1007/s11263-016-0943-0. [Online]. Available: https://doi.org/10.1007/s11263-016-0943-0.

[33] D. Lu, V. C. Jammula, S. G. Como, J. D. Wishart, Y. Chen, and Y. Yang, "CAROM - vehicle localization and traffic scene reconstruction from monocular cameras on road infrastructures," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11 725–11 731, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233004562.

[34] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," *CoRR*, vol. abs/1909.12605, 2019. arXiv: 1909.12605. [Online]. Available: http://arxiv.org/abs/1909.12605.

[35] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1–6. DOI: 10.1109/AVSS.2017.8078516.

[36] V. Eiselein, E. Bochinski, and T. Sikora, "Assessing post-detection filters for a generic pedestrian detector in a tracking-by-detection scheme," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1–6. DOI: 10.1109/AVSS.2017.8078484.

[37] T. I. Amosa, P. Sebastian, L. I. Izhar, *et al.*, "Multi-camera multi-object tracking: A review of current trends and future advances," *Neurocomputing*, vol. 552, p. 126 558, 2023, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2023.126558. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231223006811.

[38] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 2016. DOI: 10.1109/icip.2016.7533003. [Online]. Available: https://doi.org/10.1109%2Ficip.2016.7533003.

[39] H.-M. Hsu, T.-W. Huang, G. Wang, J. Cai, Z. Lei, and J.-N. Hwang, "Multi-camera tracking of vehicles based on deep features re-ID and trajectory-based camera link models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2019.

[40] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, "Exploit the connectivity: Multi-object tracking with TrackletNet," *CoRR*, vol. abs/1811.07258, 2018. arXiv: 1811.07258. [Online]. Available: http://arxiv.org/abs/1811.07258.

[41] D. Serrano, F. Net, J. A. Rodríguez, and I. Ugarte, *TrackNet: A triplet metric-based method for multi-target multi-camera vehicle tracking*, _eprint: 2205.13857, 2022.

[42] Y. Jeon, D. Q. Tran, M. Park, and S. Park, "Leveraging future trajectory prediction for multi-camera people tracking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023, pp. 5399–5408. DOI: 10.1109/CVPRW59228.2023.00570.

[43] M. S. Jazayeri, A. Jahangiri, and S. G. Machiani, "Predicting vehicle trajectories at intersections using advanced machine learning techniques," *San Diego State University;Safety through Disruption (Safe-D) University Transportation Center (UTC)*, May 1, 2021, Publisher: United States. Department of Transportation _eprint: https://doi.org/10.15787/VTT1/AKKZ6V. DOI: 10.15787/VTT1/AKKZ6V. [Online]. Available: https://doi.org/10.15787/VTT1/AKKZ6V.

[44] Q. You and H. Jiang, *Real-time 3d deep multi-camera tracking*, _eprint: 2003.11753, 2020.

[45] Y. He, X. Wei, X. Hong, W. Shi, and Y. Gong, "Multi-target multi-camera tracking by tracklet-to-target assignment," *IEEE Transactions on Image Processing*, vol. PP, Mar. 2020. DOI: 10.1109/TIP.2020.2980070.

[46] G. Mclachlan, "Mahalanobis distance," *Resonance*, vol. 4, pp. 20–26, Jun. 1999. DOI: 10.1007/BF02834632.

[47] G. Wang, Y. Wang, R. Gu, W. Hu, and J.-N. Hwang, "Split and connect: A universal tracklet booster for multi-object tracking," *IEEE Transactions on Multimedia*, vol. 25, pp. 1256–1268, 2023. DOI: 10.1109/TMM.2022.3140919.

[48] Y. Zhang, P. Sun, Y. Jiang, *et al.*, *ByteTrack: Multi-object tracking by associating every detection box*, _eprint: 2110.06864, 2022.

[49] N. D. Reddy, M. Vo, and S. G. Narasimhan, "Occlusion-net: 2d/3d occluded keypoint localization using graph networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[50] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[51] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," *CoRR*, vol. abs/1612.00496, 2016. arXiv: 1612.00496. [Online]. Available: http://arxiv.org/abs/1612.00496.

[52] W. Ding, S. Li, G. Zhang, X. Lei, and H. Qian, *Vehicle pose and shape estimation through multiple monocular vision*, _eprint: 1802.03515, 2018.

[53] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask r-CNN*, _eprint: 1703.06870, 2018.

[54] G. Brazil, G. Pons-Moll, X. Liu, and B. Schiele, "Kinematic 3d object detection in monocular video," *CoRR*, vol. abs/2007.09548, 2020. arXiv: 2007.09548. [Online]. Available: https://arxiv.org/abs/2007.09548.

[55] M. Ahrnbom, I. Persson, and M. Nilsson, "Seg2pose: Pose estimations from instance segmentation masks in one or multiple views for traffic applications," in *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications: VISAPP*, vol. 5, SciTePress, Feb. 16, 2022, pp. 777–784. DOI: 10.5220/0010777700003124.

[56] S. Li and H.-S. Yoon, "Vehicle localization in 3d world coordinates using single camera at traffic intersection," *Sensors (Basel, Switzerland)*, vol. 23, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257891001.

[57] F. Herzog, J. Chen, T. Teepe, J. Gilg, S. Hörmann, and G. Rigoll, *Synthehicle: Multi-vehicle multi-camera tracking in virtual cities*, _eprint: 2208.14167, 2022.

[58] J. Luiten, A. Osep, P. Dendorfer, *et al.*, "HOTA: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 548–578, Oct. 2020, Publisher: Springer Science and Business Media LLC, ISSN: 1573-1405. DOI: 10.1007/s11263-020-01375-2. [Online]. Available: http://dx.doi.org/10.1007/s11263-020-01375-2.