Evaluation of Multi-Camera-Based Localization for Accurate Collision Risk Detection

Maxime Cancouët, Romain Bellessort, Eric Nassor, Hervé Ruellan, Jean-Marie Bonnin

Evaluation of Multi-Camera-Based Localization for Accurate Collision Risk Detection

Maxime Cancouët
Canon Research Centre France
Rennes, France
maxime.cancouet@crf.canon.fr

Romain Bellessort
Canon Research Centre France
Rennes, France
romain.bellessort@crf.canon.fr

Eric Nassor
Canon Research Centre France
Rennes, France
eric.nassor@crf.canon.fr

Hervé Ruellan
Canon Research Centre France
Rennes, France
herve.ruellan@crf.canon.fr

Jean-Marie Bonnin
dept. SRCD
IMT Atlantique / IRISA
Rennes, France
jean-marie.bonnin@irisa.fr

***Abstract*—Accurate detection and localization of objects are key aspects of connected mobility and play pivotal roles in ensuring road safety. In particular, precise localizations are crucial for predicting potential collisions between vehicles and vulnerable road users (VRUs) crossing streets. This paper presents an evaluation of the accuracy of our camera-based roadside infrastructure in the context of collision risk detection. Specifically, our evaluation entails the deployment of two differently oriented cameras capturing a scene where a vehicle and a pedestrian converge towards a common point. By comparing camera-acquired object positions with ground truth data obtained through GNSS RTK-equipped objects, we evaluate the detection precision of our system, and we study the impact of occlusion on the results. Through this evaluation, we assess that our infrastructure achieves 60-cm positioning accuracy in real-world scenarios, providing accurate detection timing and therefore making it usable for collision detection.**

***Index Terms*—connected mobility, collision detection, near miss, positioning accuracy, cooperative perception, roadside infrastructure**

I. Introduction

In the era of Cooperative, Connected, and Automated Mobility (CCAM), roadside infrastructure equipped with sensors (e.g., camera, LiDAR, radar) and communication capabilities plays an essential role in providing precise environmental information to Intelligent Transport System Stations (ITS-S) [1]. This information, broadcasted in standardized messages like Collective Perception Messages (CPM) [2], enhances decision-making processes such as collision prediction [3] [4].

In this context, we developed a state-of-the-art perception software, the Traffic Monitoring Tool (TMT). The TMT is capable of detecting objects and extracting information from video frames, which can then be described in standard-formatted messages. Our solution is therefore deployment-ready and usable with any camera and radio communication means.

Our study evaluates the TMT collision detection performance using a dataset from the *SELF assessment, protection & healing tools for a trustworthY and resilient CCAM* (SELFY) European project[1], which includes video frames, LiDAR point clouds, and ground truth positions obtained with centimeter-level accuracy thanks to the Real-time kinematic (RTK) positioning technology [5].

In details, our evaluation consists in comparing positions computed from images captured by two cameras with the ground truth in order to validate the TMT accuracy for collision detection. In particular, our contributions are the following :

- We evaluate the positioning accuracy of the camera-based measurements for two different classes of objects.
- We evaluate the accuracy of the camera-based measurements in terms of distance between the two objects.
- Based on this distance, we evaluate the correct detection by our system of a collision risk by defining several levels of sensibility. In particular, we study the alignment of detection time of our system with the ground truth.
- We study the impact of an object occlusion on those results.
- In addition, we published the dataset we used on Zenodo[2].

In the next section we first review the literature background, including the well-known technologies and methods used in the TMT, as well as the existing methods in collision detection and roadside infrastructure evaluation. Then, we describe our evaluation methodology, including the scenario description, the hardware setup and the corrections made on positions. Finally, we describe and discuss our results in a last part, before concluding on the work done.

The SELFY project has received funding from the Horizon Europe programme under grant agreement No. 101069748. Funded by the European Union. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

[1]https://selfy-project.eu/
[2]https://zenodo.org/records/11026094

II. Related Work

A. Traffic Monitoring Tool

Our cooperative perception architecture, the TMT, is similar to other architectures [6] [7] [8]. As described on Fig. 1, it includes a Video Content Analytics (VCA) tool using a Faster R-CNN model [9] to detect and extract objects' information (e.g., localization and class) from each camera frame. The calibration of the camera is then used to estimate the positions of the detected objects in Universal Transverse Mercator (UTM) coordinates. These coordinates are then filtered thanks to a Kalman filter, allowing the object's speed computation. The process is detailed below:

- **Camera calibration**: An initial step in the configuration of the TMT is to estimate the calibration of the camera. Our calibration process is based on well-known methods [10] to compute both the intrinsic and extrinsic matrices enabling to convert positions in the image into locations using UTM coordinates. Practically, we developed a tool allowing the user to manually select corresponding points on a map and on a reference frame captured by the camera. The more points selected, the more precise the position estimation will be, especially in the farthest parts of the Field of View (FoV).
- **Object detection**: Our main tool is the Video Content Analytics (VCA) which is responsible for detecting objects in a video frame, extracting their properties like a bounding box, a position, a class, etc. It uses a modified Faster R-CNN model with a ResNet 50 backbone [11] provided by TorchVision[3]. In addition, we added an option that allows a spatial filtering in the FoV, so that objects are detected only in a specific region.
- **Position estimation**: Using the object position in the image and the calibration previously computed, we convert this position into UTM coordinates.
- **Object Tracking**: Finally, the obtained UTM coordinates are filtered using a Kalman Filter to obtain a better estimation of the objects' positions and to also estimate their speeds. [12]. We developed bounding-box tracking as well as position tracking, but we use position tracking as we noticed it is achieving better precision.

These object properties are then organized in standardized-like messages and solely need an encoding step before actual sending to other stations.

B. Previous Work

Collision detection and near miss detection are subjects of great interest in the literature.

In particular, some of the most used metrics to detect near misses are the Time to Collision (TTC) and Post Encroachment Time (PET). TTC can be defined as the estimated time left before two objects' trajectories meet. PET is the time left before one of the object has entirely passed the other object's trajectory.

[3]https://catalog.ngc.nvidia.com/orgs/nvidia/resources/resnet_50_v1_5_for_pytorch

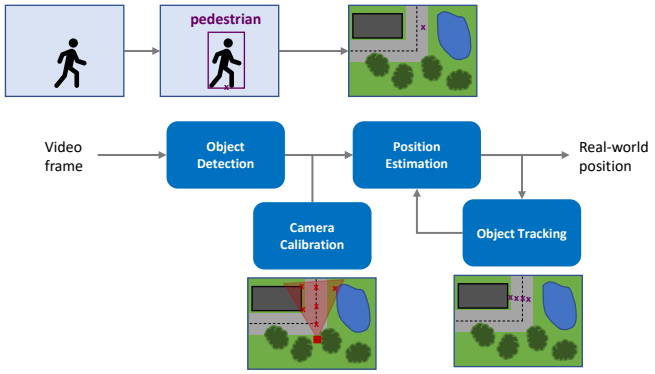


Fig. 1: TMT architecture

For example, the authors of [13] propose a distance-based method to detect a collision from a vehicular point of view. Their idea is to use a camera and object detection techniques to extract objects'bounding boxes whose sizes are then compared to the size of the FoV. This ratio is used to estimate the distance between the object and the vehicle, and if a threshold is exceeded, a collision risk is detected.

Some papers propose other techniques of collision detection. For example in [14], the authors propose a set of tools to achieve near miss detections from a road-side point of view, and evaluate their tools. Similarly to well-known methods, they process multi-camera video frames to obtain positions of detected objects and track them through time. Then, they use TTC and PET to detect and classify near misses. They propose an evaluation with the Synthehicle (synthetic data using CARLA) and FlowCube (real video) datasets. By comparing it with a ground truth automatically obtained (Synthehicle) or manually determined (FlowCube), they achieve an accuracy of respectively 0.36 m and 0.72 m. As for the near miss detections, the system is consistent with the ground truth.

This work shows well that the estimated positions are critical data, as the TTC, PET, heading and speed are derived from them. Therefore, it is mandatory to evaluate localization accuracy using a precise ground truth.

In literature, several authors evaluate the accuracy of their infrastructure with different methods. For example, Providentia [6] is an Intelligent Infrastructure System (IIS) with multiple sensors (cameras and radars) aiming at building a digital twin of a highway, for different applications. Sensors are placed on two gantry bridges so that fields of view overlap. This allows a more precise determination of the positions of the vehicles seen. Objects are tracked for each sensor and then data from the different sensors are fused in order to achieve a precise localization of the objects. The authors evaluate their IIS by comparing the localization of the objects detected by the sensors with a ground truth obtained after mapping an aerial view of the highway (taken from a helicopter) on a reference map. They achieve a Root Mean Square Error (RMSE) of 3.31 m (3.27 m in longitudinal direction and 0.53 m in lateral direction). They explain it by the fact that the position given by the algorithm usually takes the bottom center of the bounding

box of the objects, which is not the real center of the position of the vehicles.

Also, some papers propose to evaluate their infrastructure using GNSS RTK antennas. For instance, [8] describes the design of a real-time system of localization of vehicles for large distances based on camera (using Yolov3). The authors' objective is to accurately plan a lane merge maneuver. They propose an evaluation method where vehicles are equipped with GNSS RTK antennas and return a precise localization that is considered the ground truth. RTK antennas bounding boxes and initial positions are manually selected in the first video frame and then tracked thanks to a Kalman Filter. Then, the 2D image positions are projected into a plane parallel to the road, and with the measured height of the antennas they are able to reconstruct the 3D positions of the vehicles for each image. The positions of the vehicles detected by the camera are then compared to the ground truth. The evaluation shows that the difference between the camera-based and the GNSS RTK localization is in the order of 4 m for a distance of up to 160 m.

The authors of [7] also propose an implementation of a surveillance camera-based positioning system based on Yolov4, with no tracking implemented. To evaluate the accuracy of positions in their implementation, they propose an experimentation with an outdoor scene and a pedestrian in vicinity. The pedestrian has a GNSS RTK on his head which acts as the ground truth. The positions of the pedestrian computed from the camera images are compared to this ground truth to obtain the absolute error. As the distance of the pedestrian from the camera varies from 5 m to 8 m, the mean positioning error varies from 14.5 cm to 18.9 cm with a standard deviation respectively equal to 6.7 cm and 8.7 cm. After detection of a constant bias toward the camera, the authors added an offset to the vertical coordinates, reducing the mean position error to 10.7 and 15.6 cm.

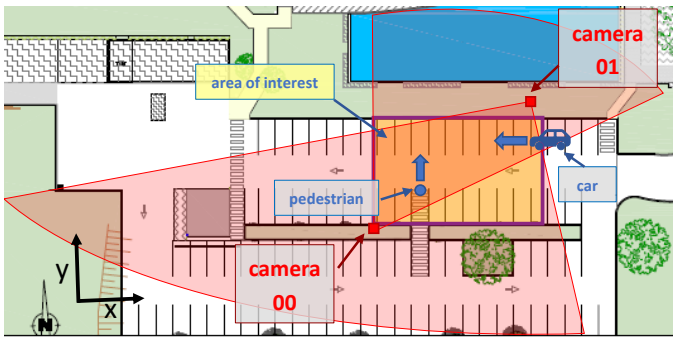
III. Evaluation Methodology

In this section, we detail our evaluation methodology. First, we set two cameras that we calibrate and connect to the TMT. Second, we define an area of interest, which is the area where we want to detect collision risks. Third, we have a pedestrian and a vehicle moving toward the same point, and therefore simulating a near miss detection. Finally, we compare the positions output by the TMT to the positions obtained through the GNSS RTK antennas installed on both the pedestrian and the vehicle, and we determine whether the collision risk can accurately be assessed by our system.

A. Scenario Description

As we can see on Fig. 2, the scene was captured in a parking lot. We set up two cameras in different locations of the parking lot but with overlapping FoVs in order to create redundancy in the estimation of the localization of the objects. Camera 00 is fixed on a lighting pole and is at a height of around 350 cm, while camera 01 is at a height of around 250 cm from the ground of the parking lot.

(a) FoV of camera 00. (b) FoV of camera 01.



(c) Map of the scene.

Fig. 2: Experimental setup.

TABLE I: Sensors used during experimentation.

Sensor	Model	Parameters
Camera 00	Axis P5635-E	frame rate: 10 Hz
Camera 01	Axis Q1656	resolution: 1920x720
GNSS (car)	ArduSimple AS- Prokit - HandSurvey	update rate: 10 Hz
GNSS (pedestrian)	ArduSimple AS- Prokit - HandSurvey	update rate: 10 Hz

In this scenario, a pedestrian is located at the south of the parking lot and walks in a straight line toward the north, while a car is located at the east of the parking lot and drives in a straight line to the west of the parking lot. On camera 00, both car and pedestrian are outside the FoV when they start moving, while on camera 01 only the car is outside the FoV at the start.

As in a real world deployment, we focused our experiment on a specific area of the FoV. To do so, the TMT supports the definition of an area of interest. Objects detected outside this area are ignored (such as far-away cars that can be seen on Fig. 2b).

At some point, the pedestrian and the car are in a near miss position, meaning the car drives just in front of the pedestrian. For the safety of the pedestrian, he stops at a line drawn on the parking lot which is at least 100 cm away from the car.

Also, from the perspective of camera 01, the pedestrian is occluded at some point, as the car drives in front of him.

Both the car and the pedestrian are equipped with GNSS RTK antennas connected to smartphones in order to record their ground truth positions while they move. The base station used for RTK position corrections is approximately 34 km away, ensuring an accuracy of a few centimeters.

Sensors used for the experiment and their main characteristics are summarized in Table I.

TABLE II: Corrections applied to positions.

Problem	Correction	
	x (cm)	*y (cm)*
Cartography service offset	-55	-57
Car antenna position	-136	None

B. Objects Positioning

To run the TMT, we used a computer equipped with a NVIDIA GeForce RTX 3090 graphic card, allowing to process a frame from each camera in less than 60 ms (16,6 Hz). It is to be noted that the time of processing is heavily impacted by the time the TMT needs to load one image from the dataset (approximately 20% of the time). As the frames are stored on a classical Hard Disk Drive (HDD), this loading time could be improved by storing the frames on a Solid-State Drive (SSD). As an indication, in another experiment in which the TMT was directly connected to a camera and performed live analysis of video frames, a frame rate of 40 Hz was achieved using an NVIDIA GeForce RTX 4090 Ti.

The positions given by both the TMT and ground truth are given using UTM coordinates (in centimeters). Therefore, we used a reference frame aligned with east (x) and north (y) directions. Also, timestamps are given in seconds and the origin date is corresponding to the beginning of the experiment.

As it is almost constant in our parking lot, we consider the altitude parameter (z) to be constant, and therefore we did not evaluate it.

In addition, after collecting and analyzing our data, it appeared that a constant offset existed between the ground truth and the selected cartography service. By measuring this offset for multiple ground truth positions, we determined that this offset corresponded to a shift of 55 cm in x and 57 cm in y directions. Please note that this offset, in itself, does not change the ability of the TMT to be used for detecting collision risk. However, as our goal is to compare TMT results to ground truth, we need to take this shift into account. Furthermore, as the antenna position was at the rear of the vehicle (and not in the center), it was necessary to offset the ground truth positions of the car of 136 cm mostly in the x direction. Again, this shift is necessary only so that TMT positions and GNSS RTK positions can be compared. It has no impact on the TMT itself. The corrections are summarized in Table II.

To evaluate the accuracy of our system, we compute the euclidean distance between our camera-based positions and the ground truth as the position error. Also, in the context of a collision detection scenario, we model the shape of the vehicle around the positions and model a safety zone around the pedestrian. We then compute the distance between these two areas for each source and each timestamp. Finally, as timing is crucial in a collision risk scenario, we verify if the detection of potential collisions using camera measurements is aligned with the detection of potential collisions using the ground truth.

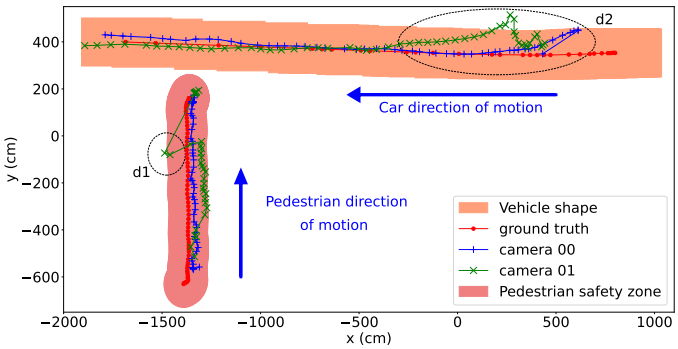


Fig. 3: x and y positions of car and pedestrian.

IV. Results and Discussion

In this section we review and discuss the results of different aspects of the experimentation.

A. Results

On Fig. 3 we can see the different trajectories of the car and the pedestrian computed from the camera images and measured as the ground truth. We modeled the vehicle as a rectangle (with the same size as the vehicle) and the pedestrian safety zone as a disk of 100 cm radius. These areas are represented around each point of the ground truth positions. As planned during the experiment, the vehicle never enters the pedestrian safety zone.

Still on Fig. 3, we can see two main differences (named $d1$ and $d2$ on the figure) between the positions computed from camera images and the ground truth:

- The first difference $d1$ is for the positions of the pedestrian as computed from camera 01 frames: this is due to the occlusion of the pedestrian by the car. As the car is driving in front of the pedestrian, only partial detections of the pedestrian are possible, leading to an erroneous estimation of his position, which is shifted to the left of the x-axis. However, we see that the pedestrian last positions (after the occlusion) are again aligned with the ground truth.
- The second difference $d2$ is for the positions of the car as computed from both camera frames: while the cameras see only a part of the car, the position estimation module has difficulties estimating correctly its position, and the tracking module struggles at predicting its speed, heading and position. Furthermore, while the car is entering the FoV of the cameras, its shape is distorted by the cameras lenses. These factors are causing this “FoV border effect”, which results in a decreased accuracy at the borders of the FoV.

As the detections on the edges of the FoV have a great impact on the results, we calculate the euclidean distance in two cases: first, for the full trajectories; second, for the nominal situation, where the nominal situation corresponds to the instants when the objects are fully seen by the camera. Practically, we compute the euclidean distance in the nominal

TABLE III: Euclidean position error e to the ground truth and standard deviation σ of the error.

Sensor	Object	Full track		Nominal track	
		e (cm)	σ (cm)	e (cm)	σ (cm)
camera 00	car	42.81	47.02	21.60	13.65
	pedestrian	32.64	11.11	32.64	11.11
	All	**37.54**	**33.84**	**29.13**	**12.96**
camera 01	car	89.14	62.05	49.04	41.24
	pedestrian	69.72	30.30	69.72	30.30
	All	**81.42**	**52.45**	**60.18**	**36.97**
System	**All**	**56.85**	**48.15**	**42.59**	**30.33**

situation by removing the points where the objects are not fully seen. In this scenario, only the car is affected. The results can be seen in Table III.

We see that for both sensors, the pedestrian positions are more accurate than the car positions. By calculating for each sensor the position errors for both objects for the full track, we get a mean error of 37.54 cm for camera 00 and 81.42 cm for camera 01. Overall, camera 00 has a lower error than camera 01. A first reason is that lens distortion are greater for camera 01 than for camera 00. In addition, the locations of the car and the pedestrian inside the FoV of camera 01 makes it harder to correctly compute their positions. Also, the pedestrian is far away from the camera, while the car is very close to the camera at the beginning. We also see that the mean standard deviation is high for the car on both cameras, mostly due to the FoV border effect. In particular, the results for the car vary significantly between the full track and the nominal situation. Finally, the overall system error is greatly reduced in the nominal situation, decreasing from 56.85 cm in the full track to 42.59 cm, while the standard deviation is decreased by almost a factor 1.5, due to the border effect when the car is not completely in the FoV.

As our purpose is to detect risks of collision, we define $dist_{PC}(t)$ as the shortest distance between the car shape and the pedestrian safety zone. The distance according to the ground truth is defined as $dist_{PC}^{GT}(t)$ while $dist_{PC}^{C0}(t)$ and $dist_{PC}^{C1}(t)$ respectively correspond to the distance according to camera 00 and camera 01. The results are shown on Fig. 4.

Similarly to [13], in order to characterize the ability of the system to detect risks of collision with an accurate timing, we define d as the distance below which a risk should be detected. For different values of d, we compare the amount of time when a risk is detected respectively by the ground truth and by our system. As $min(dist_{PC}^{GT}(t)) = 31\ cm$, and as our system error is near 60 cm, we have d to vary from 30 to 90 cm by 10 cm increment. In addition, as the two cameras are meant to be used together, we consider that a risk of detection is identified by our system as soon as a risk is detected by one of the cameras.

The results are shown in Table IV, with the duration of both true positive and false positive detections by our system. As d increases, our system consistency with the ground truth increases. In particular, and as illustrated by Fig. 5, in the

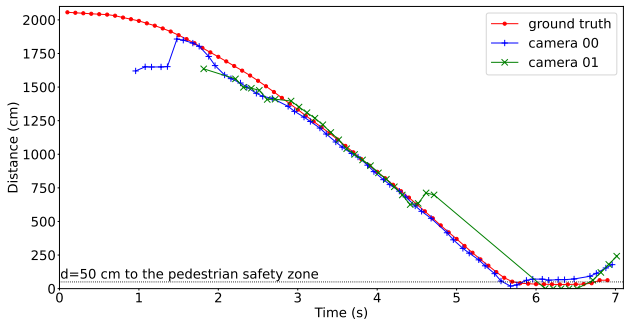


Fig. 4: Distance between the pedestrian safety zone and the car depending on time.

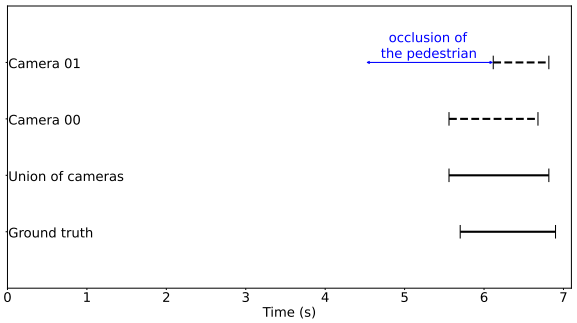


Fig. 5: Time intervals where a collision risk is detected for d=80 cm.

case where $d = 80\ cm$, our system is mostly aligned with the ground truth, with a detection duration of 1.26 s for our system and 1.20 s for the ground truth, our system identifying a risk 0.14 s before the ground truth. As one may see on Fig. 5, camera 01 is deeply impacted by the occlusion of the pedestrian, but our system still manages to identify the risk thanks to the presence of camera 00. This shows the benefits of having two cameras recording the scene from two different angles.

B. Discussion

With a simple combination of camera 00 and camera 01, we see that we are able to minimize the errors and therefore keep a minimal difference of detection time between the system and

TABLE IV: Risk detection duration of ground truth and TMT with d varying.

d (cm)	Duration risk ground truth (s)	Duration true positive TMT (s)	Duration false positive TMT (s)
30	0.00	0.00	0.76
40	0.80	0.58	0.02
50	1.00	0.68	0.12
60	1.10	0.78	0.14
70	1.20	0.96	0.14
80	1.20	1.12	0.14
90	1.30	1.22	0.04

the ground truth. Nevertheless, this could be further enhanced with the data fusion of the results of both cameras.

Also, as this first experiment shows that our system assessment of collision risk is consistent with the ground truth, it means that it could be used for more advanced scenarios, especially with more objects. Running such scenarios would be helpful to better understand the potential limitations of our solution. In particular, determining collision risks for objects with close trajectories may be more challenging.

V. Conclusions

This paper provides an evaluation of our perception infrastructure software, the Traffic Monitoring Tool (TMT), for detecting collision risk. We used the SELFY dataset which allows a real-world evaluation thanks to its accurate ground truth obtained with the GNSS RTK technology as well as its video frames obtained with two differently oriented cameras. We evaluated our system to have an accuracy of about 60 cm in the full FoV of the cameras, which is improved to about 40 cm in the nominal situation where the objects not fully in the FoV are not considered. Furthermore, we also evaluated the estimation of the collision risk using the TMT measurements. The results show that the TMT assessment of collision risk is consistent with the ground truth, which validates that it can be used in such use cases, even in situations where occlusions may occur.

Nevertheless, some improvements are possible. The positioning accuracy could in particular be improved at the border of the FoV, as well as by fusion on the data obtained from the two cameras.

Also, more advanced testing using more of the objects properties (i.e., the speed and heading in addition to the position and shape) in the dataset could be achieved to have a better estimation of the collision risk detection. For this purpose and for others, we published the dataset we used on Zenodo.

Finally, to determine whether our approach may work in more challenging situations, we may also investigate in a future work more complex scenarios, where the number of objects as well as their trajectories are more diversified.

Acknowledgment

This work has been performed in the scope of the SELFY Project funded by the Horizon Europe programme under grant agreement No. 101069748. The authors would like to acknowledge the contributions of their colleagues from YoGoKo who participated in the experimentation.

References

[1] M. Correia, J. Almeida, P. C. Bartolomeu, J. A. Fonseca, and J. Ferreira, "Performance Assessment of Collective Perception Service Supported by the Roadside Infrastructure," *Electronics*, vol. 11, no. 3, p. 347, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/3/347

[2] *ETSI TS 103 324 - V2.1.1*, ETSI ITS Std., 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103300_103399/103324/02.01.01_60/ts_103324v020101p.pdf

[3] T. Gandhi and M. Trivedi, "Pedestrian collision avoidance systems: A survey of computer vision based recent studies," in *2006 IEEE Intelligent Transportation Systems Conference*. IEEE, 2006, pp. 976–981. [Online]. Available: http://ieeexplore.ieee.org/document/1706871/

[4] Y. Huang, Y. Wang, X. Yan, X. Li, K. Duan, and Q. Xue, "Using a V2V- and V2I-based collision warning system to improve vehicle interaction at unsignalized intersections," *Journal of Safety Research*, vol. 83, pp. 282–293, 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0022437522001384

[5] A. El-Mowafy, "Precise Real-Time Positioning Using Network RTK," in *Global Navigation Satellite Systems: Signal, Theory and Applications*, S. Jin, Ed. InTech, 2012. [Online]. Available: http://www.intechopen.com/books/global-navigation-satellite-systems-signal-theory-and-applications/precise-real-time-positioning-using-network-rtk

[6] A. Krämmer, C. Schöller, D. Gulati, V. Lakshminarasimhan, F. Kurz, D. Rosenbaum, C. Lenz, and A. Knoll, "Providentia – A Large-Scale Sensor System for the Assistance of Autonomous Vehicles and Its Evaluation," *Journal of Field Robotics*, vol. 2, no. 1, pp. 1156–1176, 2022. [Online]. Available: https://fieldrobotics.net/Field_Robotics/Volume_2_files/Vol2_38.pdf

[7] T. Partanen, P. Muller, J. Collin, and J. Bjorklund, "Implementation and Accuracy Evaluation of Fixed Camera-Based Object Positioning System Employing CNN-Detector," in *2021 9th European Workshop on Visual Information Processing (EUVIP)*. IEEE, 2021, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9483987/

[8] K. Cordes, N. Nolte, N. Meine, and H. Broszio, "Accuracy Evaluation of Camera-based Vehicle Localization," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 2019, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/8965230/

[9] S. Ren, K. He, R. Girshick, and J. Sun. (2016) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. [Online]. Available: http://arxiv.org/abs/1506.01497

[10] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov./2000. [Online]. Available: http://ieeexplore.ieee.org/document/888718/

[11] K. He, X. Zhang, S. Ren, and J. Sun. (2015) Deep Residual Learning for Image Recognition. [Online]. Available: http://arxiv.org/abs/1512.03385

[12] M. M. Rana, N. Halim, M. M. Rahamna, and A. Abdelhadi, "Position and Velocity Estimations of 2D-Moving Object Using Kalman Filter: Literature Review," in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2020, pp. 541–544. [Online]. Available: https://ieeexplore.ieee.org/document/9061241/

[13] S. Saleh, C. Rellan, and S. P. Surana, "Collision Warning Based on Multi-Object Detection and Distance Estimation," in *2020 International Symposium on Computer Science, Computer Engineering and Educational Technology*, 2020. [Online]. Available: https://www.researchgate.net/publication/348155370_Collision_Warning_Based_on_Multi-Object_Detection_and_Distance_Estimation

[14] L. van der Weide, "Near-miss detection on traffic intersections with a distributed overlapping multi-camera system," April 2024. [Online]. Available: http://essay.utwente.nl/98720/