IEEE International Conference on Intelligent Transportation Systems (ITSC), 18 Sep. - 12 Oct. 2022.

Infra2Go: A Mobile Development Platform for Connected, Cooperative and Autonomous Driving

Tobias Fleck[1], Lennart Jauernig[1], Rupert Polley[1], Philip Schörner[1], Marc René Zofka[1] and J. Marius Zöllner[1,2]

***Abstract*— Connected, cooperative autonomous driving and mobility promises increased comfort and safety for public transportation and logistics in urban and suburban regions. Stationary roadside infrastructure equipped with intelligent perception sensors and communication units has the potential to increase the field of view and mitigate occlusions in perception, but is limited to fixed places in the development process.**

We present a mobile development platform for connected roadside infrastructure that can be used in an extensible and flexible fashion for various purposes: as a traffic monitoring device, as a reference sensor platform to benchmark in-vehicle perception, as a test platform for collaborative perception algorithms and to operate as connected edge computation infrastructure to support vehicles in decision making in real world settings.

In this work, we present an overview over the Infra2Go concept and give details about the hardware and software architecture. We evaluate our platform by deploying it in an outdoor real world scenario for cooperative perception between the platform and a vehicle equipped with an onboard V2X communication unit. The platform is used to track vulnerable road users (VRU) and transmits the information via V2X communication for cooperative perception (Cooperative Perception Messages) to an autonomous vehicle in order to extend it's sensor field of view.

I. INTRODUCTION

Connected, cooperative and automated mobility (CCAM) has been a topic of interest for several years. Multiple national and international activities and research projects have been carried out showing early feasibility and potentials. Early applications of infrastructure supported driving and infrastructure based warning services [1] have been demonstrated in urban scenarios. Gabb et. al. [2] demonstrated that cooperative perception by track level fusion between infrastructure and autonomous vehicles is possible and can improve data quality. Shan et. al. [3] demonstrated collaborative perception using a mobile platform based on LiDAR and Camera. Tarko et al. [4] introduced TScan a mobile platform based on a transporter with a telescopic mast system that relies on LiDAR, inertial sensors and camera data to track road participants.

Besides technical feasibility studies, major normative efforts for the connected driving domain exist. In the Inframix project [5], connectivity and levels of infrastructure capabilities are normed by different support classes. A crucial part in standardization and norming is done by the *European Telecommunications Standards Institute (ETSI)* which standardizes communication protocols and messages for *Vehicle-to-X communication (V2X)* to exchange data in connected vehicles like *Cooperative Awareness Messages (CAM)* [6], *Cooperative Perception Messages (CPM)* [7] or *Vulnerable Road User Awareness Messages (VAM)* [8] that are promising to improve safety of *vulnerable road users (VRU)* by considering external information in automated and manually driven vehicles. These standardization activities emphasize the importance of the topic of infrastructure based VRU perception, communication and connected driving in general.

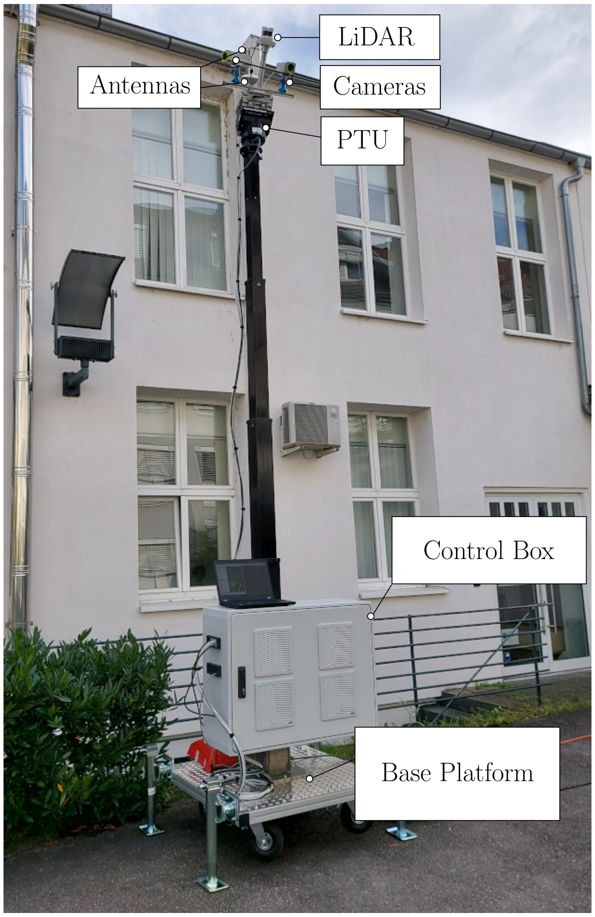


Fig. 1: The Infra2Go mobile platform that is designed as a testing platform for infrastructure based autonomous driving research and as a reference sensor platform.

[1] FZI Research Center for Information Technology, 76131 Karlsruhe, Germany. {tfleck, polley, schoerner, zofka, zoellner}@fzi.de, lennartjauernig@outlook.de

[2] Karlsruhe Institute of Technology (KIT), Germany. marius.zoellner@kit.edu

Multiple research activities construct or use mobile roadside infrastructure as a testbed for connected driving application. Liu et. al. [9] build a mobile edge platform that relies on camera data and local computation and preprocessing. Kahn et. al. [10] introduce a platform that relies on camera and infrared camera data.

Static roadside infrastructure is more commonly used. Our previous efforts in the Test Area Autonomous Driving Baden-Württemberg (TAF-BW) put the focus on stationary intelligent roadside infrastructure [11] and real time multi-object tracking [12] in static roadside infrastructure. Similar activities can be observed in many other test sites [13–17], just to mention a few. For more information on stationary roadside infrastructure for reference data acquisition, Hoss et al. [18] gives an overview. A survey on intelligent transportation with external infrastructure is given in [19].

None of the presented approaches provides a feasible and suitable solution for the design and conceptualization of a flexible infrastructure platform. Hence we identify the need for a mobile, modular testing and development platform that supports flexible testing of infrastructure and cooperative algorithms for infrastructure based autonomous driving under real world conditions. This may not be limited by sharing perception information between the infrastructure and the vehicle but may also include more advanced information sharing. For instance, trajectory planning or other high level tasks may also be performed by intelligent infrastructure [20, 21]. The platform is required to support such testing activities.

II. USE CASES AND REQUIREMENTS

We identify the following use cases for the mobile platform:

A. Data recording

The development of new technologies and algorithms strongly depends on reference data. Especially machine learning techniques require a vast amount of diverse data. The platform therefore needs to serve as a flexible instrument to record data of various sensor types and from arbitrary locations and perspectives.

B. Data processing

Besides data recording, the platform needs to be able to process data on the spot. This might be necessary to preprocess gathered data, e.g. to ensure privacy, to evaluate developed concepts in the field or to run applications based on the perceived data in order to pass on only processed and condensed information. As the range of applications is countless, it has to be ensured that new concept, algorithms and processing pipelines can be deployed easily. The computing hardware also needs to be suitable to meet the demands of the application, e.g. for machine learning based applications.

C. Data transmission

One of the aforementioned applications is the extraction and broadcasting of situation aspects of the traffic environment. In the context of automated driving specific message protocols have been defined by the ETSI ITS-G5 CP Service [7]. The objects in the environment are perceived and the observations are transmitted to other traffic participants via V2X communication.

D. Environment

Besides the targeted use cases, we pose further requirements. First of all, we want to use the platform in all weather conditions, if it's raining or if it is exposed to the sun in summer. Hence, the electronic components need to be stored in a waterproof case equipped with an active cooling system. Moreover, it is necessary to be able to reach a certain height with the sensors to cover a large area and to conveniently simulate static roadside infrastructure. Easy in height adjustment of sensor orientation gives flexibility when testing different sensors and sensor modalities. Hence, to be able to adjust the sensor orientation without retracting and extending the mast, the sensor head needs to be electronically controllable. Last, the platform needs to be movable for transportation and has to be easily adjustable at the target location. When the platform is located at the desired spot, it must be possible to set up the platform safely.

III. INFRA2GO - HARDWARE CONCEPT

In the following, the overall hardware components of the platform are described to give an overview of its components, which are shown in Fig. 1. The platform consists of three major subsystems: a base platform provides stability with flexible pedestals to be used in uneven terrain. A telescopic mast that can reach heights of up to 4 meters is used as a mounting point for a control box containing all necessary electronics and compute hardware. A sensor head is mounted on top of the telescopic mast with a Pan-and-Tilt-Unit (PTU) that allows to move the sensor head remotely via software. All subsystems are mechanically loosely coupled to make it easy to unmount and transport the platform. The sensor head can be switched if necessary for the test scenario, e.g. to test different sensor settings or to test different kind of infrastructure sensors. All in all, the platform can reach a height of up to 5.5 meters, including the base platform, telescopic mast and sensor head.

Fig. 2 shows the electronic components that are included in the mobile platform. The core component of the platform is an electric control unit (ECU) that contains a suiting central processing unit (CPU) and a graphics processing unit (GPU) that supports Deep Learning inference using recent DL-frameworks. Hot-swap NVME solid state drives make an easy handling of local data recordings possible. A network switch that supports Power-Over-Ethernet (POE) connections enables the powering of sensors on the sensor mast. GNSS-Antennas (D-GPS), as well as a standard wifi antenna and a V2X communication unit with suiting antenna for IEEE 802.11p and/or cellular V2X (C-ITS) are included in the platform to provide connectivity. The GNSS device is able to receive correction data via the German SAPOS service [22] which makes highly accurate globale registration of the sensor data possible. To power all components, the platform

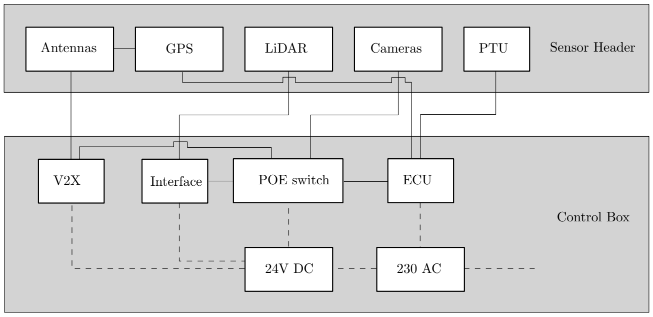
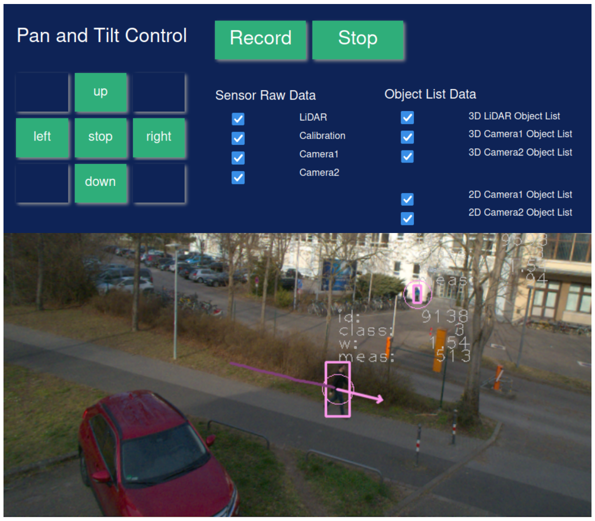


Fig. 2: The electronic components of the sensor platform.



Sensor Raw Data

☑ LiDAR
☑ Calibration
☑ Camera1
☑ Camera2

Object List Data

☑ 3D LiDAR Object List
☑ 3D Camera1 Object List
☑ 3D Camera2 Object List
☑ 2D Camera1 Object List
☑ 2D Camera2 Object List

Fig. 3: Web based user interface to move the pan-and-tilt unit, trigger data recordings and to view outputs of the algorithms on the platform.

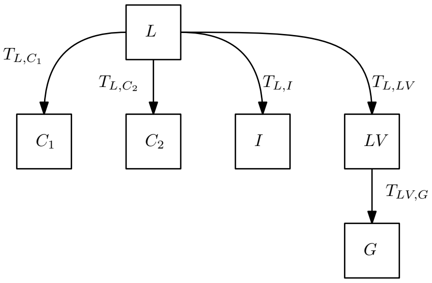


Fig. 4: Coordinate systems of the platform.

contains two power systems, a 230V AC system and a 24V DC converter to provide flexibility when integrating new components if needed. For power consumption, we focus on flexibility rather than energy optimality. An active fan based cooling system in the control box ensure stable temperatures in warm weather conditions.

The current sensor head includes a 3D-LiDAR (Ouster OS-1 with 64 layers) with integrated Interial Measurement Unit (IMU) that provides pointcloud streams of either 10Hz or 20Hz and RGB color cameras (Basler a2A1920-51gcPRO) that support frame rates of up to 51 frames per second.

IV. Infra2Go - SOFTWARE CONCEPT

For easy software integration, we rely on the Robot Operating System (ROS) as a middleware running on a standard linux operating system on the main ECU, supporting also an easy integration of common machine learning frameworks.

A. User Interface

To provide easy and flexible sensor positioning in field and measurement campaigns, a user friendly web based user interface (UI) (see Fig. 3) has been developed. The interface can be reached as a local web service when a user is connected to the platform via mobile phone or notebook. The web services can be used to query current live image streams and visualization images that show results of the tracking and detection algorithms running locally on the platform. The UI can be used to move the pan-and-tilt unit to conveniently align the sensor head for testing. Furthermore, data recordings can be triggered with the UI (see Fig. 3).

B. Time Synchronization

In order to support distributed multi sensor fusion and to make the gathered data comparable to in-vehicle data, highly accurate time stamping is needed. We utilize GPS time as a reference time source, as GPS sensors are widely available and are commonly used for time synchronization.

The main ECU is synchronized against GPS time, by using the GPS device on the platform as a reference clock. The LiDAR scanner is synchronized to the main ECU by using the PTP protocol, where the ECU acts as a master source for the LiDAR scanner, making it possible to get highly accurate time stamps on the LiDAR data. Camera data is synchronized via software by writing the current main ECU system time stamp to the data when the images are received on the host ECU. Although not being perfect, this is sufficient for most of our use cases due to the high frame rates of the cameras.

C. Spatial Calibration and Coordinate Systems

Fig. 4 shows the coordinate systems and their spatial relationships as $6DOF$ transformations $T = (t, r)$ ($3D$ translation t and $3D$ orientation r) that we define in the mobile platform. Frame L represents the LiDAR origin coordinate system, being fixed to the orientation of the LiDAR hardware. Frame I is the coordinate system of the integrated IMU of the LiDAR scanner. Coordinate systems C_i represent camera base coordinate systems for a pinhole camera model in $3D$ space. In these systems, camera i is located in the origin of C_i and the z axis is pointing towards the view direction of the camera. Coordinate system G is the ground base point of the mobile platform. LV is a virtual LiDAR coordinate system in the origin of L, but oriented similar to the ground system G.

Calibration of the mobile platform is equal to estimate the transformations T_{L,C_1}, T_{L,C_2}, $T_{L,I}$, $T_{L,LV}$ and $T_{LV,G}$. In our case, the transform $T_{L,I}$ is given by the hardware spec of the sensor.

To estimate the transforms T_{L,C_1} and T_{L,C_2}, lidar-to-camera calibration needs to be performed. We tackle this problem with a target based calibration procedure, in which a checkerboard pattern is detected in both sensor streams

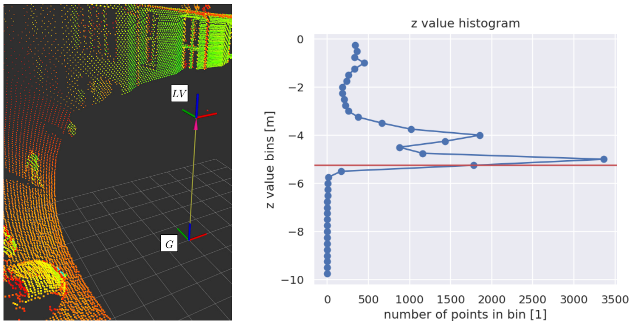


Fig. 5: LiDAR pointcloud from a parking lot with coordinate systems LV and G and the corresponding z-value histogram. The red line represents the estimated approximate height.

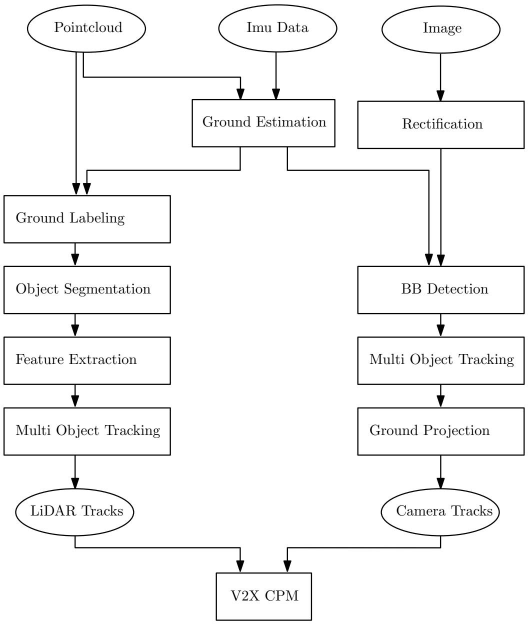


Fig. 6: Overview of the multi-object tracking pipeline.

(LiDAR and rectified camera images) and correspondence points between LiDAR and rectified camera are stored over time. Then a standard solver for the *perspective-n-point problem (PnP)* [23] is used to estimate the transform.

The intrinsic calibration and estimation of distortion parameters of the cameras is performed offline by using a checkerboard as a target with standard calibration techniques like Zhang's method [24].

The transform $T_{L,LV}$ is given by a simple rotation, as no translation between the two coordinate frames L and LV is defined. The rotation can be computed by utilizing acceleration data of the IMU. The force vector of the IMU on the static pole will point towards the direction of gravity, which is approximately equal to the direction of the ground. Hence the orientations can be computed from the force vector of the IMU data. We smooth the orientation in quaternion space by using a weighted average over a fixed window of timesteps. This procedure eliminates noise and gives a simple but stable estimate of the sensor head orientation of coordinate frame I towards the ground.

To estimate the transform $T_{LV,G}$, the height of the virtual LiDAR coordinate system towards ground has to be estimated. To do so, a simple histogram heuristic can be used. In common outdoor scenes, many LiDAR measurements and points will lie on the ground when they don't hit any objects. We utilize this observation and transform the measurement points to the coordinate system LV. In this frame, which is parallel to the ground, a histogram of the z-values (vertical axis) of points will give a peak for all points lying on the ground, if the ground is sufficiently planar. Fig. 5 shows a z-value histogram in frame LV that has been recorded in a real world scene. The heuristic starts from the lowest possible z-value (in Fig. 5 -10 meters) and returns the height where the number of points in a bin exceeds a threshold τ.

After all transformations are known, it is possible to spatially relate all data from the sensors.

V. Perception and Localization of Vulnerable Road Users (VRUs)

We integrate a baseline pipeline, see Fig. 6, for multi-object tracking (MOT) of pedestrians as they are currently a main focus of our research. Although the details of the pipeline is not in the focus of this work, we give an overview of the integrated components in the following. The pipeline consists of two independent multi-object tracking systems, one relying on LiDAR data, the second one relying on image detections and tracks that are projected to a ground plane model to receive locations in three-dimensional Cartesian space.

A. Camera Perception

Once data is received from a camera, the gathered images are undistorted using distortion parameters from the intrinsic calibration process of a pinhole camera model. The rectified image is then used in the realtime capable object detector YOLO-V4 [25] with 2D non-maximum suppression (NMS) as a postprocessing step. This yields a list of bounding boxes $b^i = (x_1^i, y_1^i, x_2^i, y_2^i, c^i, s^i)$ in the image plane, where x_1^i, y_1^i is the top left corner point of an axis aligned bounding box, x_2^i, y_2^i is the bottom right corner point, c^i is the classification of the object and s^i is an existence score for the object.

A $2D$ multi-object tracking algorithm consisting of a *Labeled Gaussian Mixture Hypothesis Density Filter (Labeled GMPHD)* [26, 27] is applied in image space considering bounding box detections as an input and giving tracked bounding boxes of the form $\hat{b}^i = (\hat{x}_1^i, \hat{y}_1^i, \hat{x}_2^i, \hat{y}_2^i, \hat{c}^i, \hat{w}^i, \hat{l}^i)$, where $\hat{x}_1^i, \hat{x}_2^i, \hat{y}_1^i, \hat{y}_2^i$ and $\hat{c}^i$ are completely similar defined to the detection case, $\hat{w}^i$ is a measure for object existence and $\hat{l}^i$ is the unique label assigned to the tracked box.

The tracked $2D$ bounding boxes are then projected to the previously estimated ground plane model as illustrated in Fig. 7. For each bounding box $\hat{b}^i$, the bottom middle point $\hat{mb}^i$ is projected using a parametric line $\hat{lb}^i$ that is intersected with the ground plane model, giving the ground point $\hat{pb}^i$ of the vulnerable road user in the ground coordinate system G. The parametric line can be computed using the

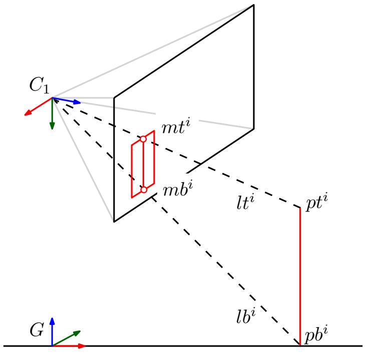


Fig. 7: Projection of tracked $2D$ bounding boxes to the ground plane model.

intrinsic parameters parameters of the calibrated camera and the transformation between camera coordinate system C_1 and the Ground G (extrinsic parameters). The top middle point $\hat{mt}^i$ of the tracked bounding box can be projected in a similar fashion by considering the intercept theorem or by using a second parametric line $\hat{lt}^i$, including the constraint that the line (pt^i, pb^i) needs to be orthogonal to the ground plane. As a result, the position $\hat{pb}^i$ and the height $||\hat{pt}^i - \hat{pb}^i||$ of a vulnerable road user can be determined. Note that this heuristic assumes that the ground point of the bounding box is standing on the ground plane model. If this assumption is not met, for instance due to partial occlusion of the bounding box or jumping of the pedestrian, positioning errors after the projection will be the result.

B. *LiDAR Perception*

The LiDAR perception pipeline uses a segmentation based approach for pedestrian tracking. As a first step, the point-cloud is ordered into a cylindrical, virtual image around the origin of the LiDAR scanner that is dependent on the resolution of the LiDAR, in our case 64×1024 pixels. Each pixel in this image refers to a single scan point in $3D$ space. The image contains different channels, including x, y, z and $range$ values of the original point, as well as $label$ and $cluster$ values that encode which classification the pixel has and to which other pixels it belongs. Label and cluster channels are initialized with default values at the beginning of the pipeline. The structure of the data is preserved through the complete pipeline.

In the second stage, labels get assigned to points, marking ground points and non-ground points in the image-like structure. This is achieved by a singular value decomposition based ground plane fitting algorithm highly inspired by [28]. The algorithm separates the space around the pole into different regions where for each region a ground hyperplane is fitted to the measurement points using an iterative scheme very similar to *Expectation Maximization (EM)*. Given a set of starting points that are assumed to be ground points, the algorithm fits a hyperplane to these points using a singular value decomposition of the empiric covariance matrix of the points, leading to a first ground estimate. Then, a reassigning step labels all points that are close to the current ground estimate to be ground points as well. This procedure is iterated by recomputing the current ground plane estimate. After a fixed number of iterations of ground estimation and reassignment, a stable ground plane is reached giving a final ground hyperplane. Finally, all points that lie within a threshold distance to the final ground estimate are labeled to be ground points, while all other points are labeled to be non-ground points. The algorithm needs a good starting hypothesis, which in our case is given by the histogram based heuristic previously described. As the height of the pole is approximately known by the histogram technique, an initial guess for the hyperplane can be set to the estimated base of the platform in parallel to the virtual lidar coordinate system that is orthogonal to the force vector of the IMU.

The third stage of the LiDAR pipeline consists of an object clustering stage. Since ground points have been labeled, but not removed, the virtual image structure on the point cloud has been preserved, which makes the use of a range image based object segmentation possible. We utilize an approach similar to FLIC [29], a fast range based segmentation algorithm that provides more stability and robustness to oversegmentation than similar approaches like [30, 31]. The third stage yields point clusters $(x, y, z, label, cluster, range)$ where identical $cluster$ values encode belonging to the same cluster.

In a fourth stage, the mean of all clusters points is computed as a tracking feature for each cluster. In a final tracking stage, these measurements are given to a Labeled GMPHD Tracker that estimates positions and velocities of the moving point objects from given position measurements, assignes unique ids over time and eliminates clutter detections, leading to a LiDAR based object list that is independent of the camera pipeline.

VI. Evaluation

To evaluate our concept, we start with a qualitative discussion of the identified requirements of the platform and their realization. We then proceed to evaluate the introduced base pipelines for VRU Localization and tracking by deploying them to the platform in a field experiment.

A. *Discussion*

Since our platform consists of three mechanical subsystems, it provides the flexibility to attach new sensor heads and to be dismounted to be transported in an ordinary transporter vehicle. Local data recordings of raw sensor data are possible with the necessary recording speeds and a convenient user interface has been designed and implemented using web services for easy access in a measurement campaign in the field. Trajectory data can be recorded from our base pipelines. The presented concept platform is weather resistant and the local cooling system ensures operation in higher temperatures. Thus all the stated requirements for flexible use are fulfilled.

B. *Vulnerable Road User Localization*

To evaluate the platform and it's capability to compute realtime trajectories of vulnerable road users, we deploy

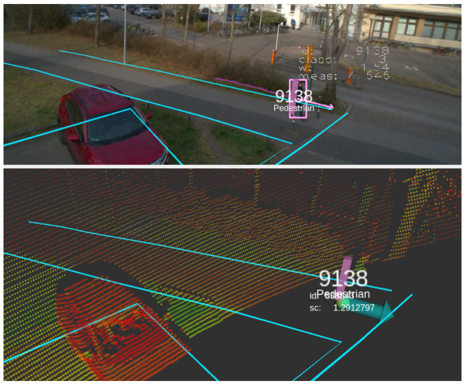


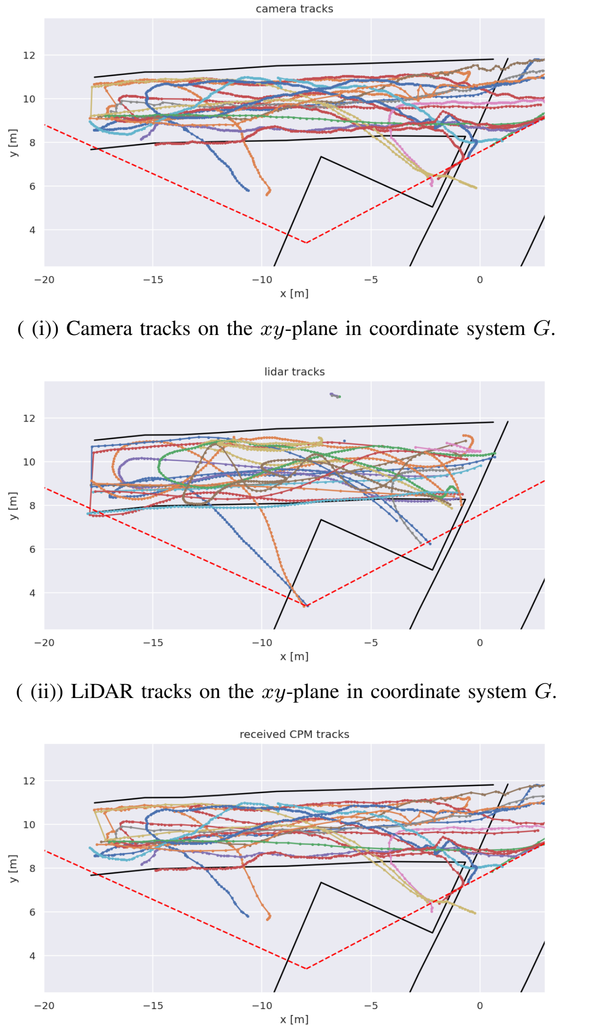
Fig. 8: Evaluation scenario from the perspective of the mobile platform. Pedestrians are tracked and located using the camera and LiDAR pipeline. The top image shows the lane network projected to the camera (cyan), the tracked 2D bounding box and the projection to the image of the corresponding 3D location of the camera track. The bottom image shows the LiDAR track as a yellow dot and it's estimated velocity vector as a cyan arrow. The text contains meta information of the tracks.

the platform in a real world traffic scenario on a walkway lane. Pedestrians walk on the walking/bicycle lane and their trajectories are estimated in real time on the platform. Fig. 8 illustrates the evaluation setting from the sensor perspective of the mobile platform.

We visualize the trajectories of a three minute recorded sequence together with a geo-referenced high definition lanelet map [32]. The mobile platform is calibrated against the map using manual correspondence matchings between the camera image and the lanelet map as described in our previous work [11]. As Figures 9 (i) and 9 (ii) visualize, both tracking pipelines produce trajectories that reach lane-precision. It can be observed that in general the estimated LiDAR trajectories are smoother and do not contain any projection errors and jumps that are cause by estimation errors of the bounding box size in the camera pipeline due to partial occlusion. Nevertheless if the explained conditions are met, camera based VRU tracking is a valid alternative to expensive LiDAR sensors. Additional occlusion reasoning might improve trajectory quality in the future.

1) Runtime Evaluation: As realtime capability is critical for perception in road user safety, we determine the distribution of pipeline delays for the two base pipelines. We do so by computing the end-to-end difference $t_{track} - t_{acq}$ between the time of the sensor data acquisition t_{acq} and the system time t_{track} when the final tracked object list is available when the corresponding measurement from time t_{acq} has been processed. Fig. 10 visualizes the distribution as histograms with confidence bounds of the results.

The camera pipeline produces object lists with a mean delay of $\mu = 120.74ms$, where the standard deviation is $\sigma = 11.72ms$. The LiDAR pipeline produces object lists with a mean delay of $\mu = 346.91ms$, where the standard deviation is $\sigma = 25.57ms$. The presented LiDAR pipeline is not optimized, while the camera pipeline relies on end-to-end detection highly leveraging the GPU computation capabilities of our platform, which explains the lower end-to-end computation times. As our LiDAR pipeline is implemented as modular components using the Robot Operating System, communication overheads between the components may introduce delays in the LiDAR pipeline that can be mitigated in the future.



((i)) Camera tracks on the xy-plane in coordinate system G.

((ii)) LiDAR tracks on the xy-plane in coordinate system G.

((iii)) CPM tracks (camera) on the xy-plane in coordinate system G that are received by the vehicle.

Fig. 9: Resulting trajectories (camera, lidar, received CPM) in the estimated xy-plane of the ground coordinate system G. Black lines show the high definition lanelet map, red dotted lines visualize the camera view frustrum.

All in all, we conclude that the mobile platform provides a flexible and powerful tool to test realtime sensor perception pipelines in real life conditions in a convenient fashion. Deep Learning based detectors and algorithms can easily be executed and tested on the platform.

C. *Cooperative, Connected Perception via V2X*

As an evaluation of the concept of infrastructure supported sensing of vulnerable road users, we integrate our camera tracking pipeline with the V2X communication hardware of the platform. The tracked object lists generated by the camera are converted to CPMs with a custom message sampling procedure that selects CPMs uniformly with a fixed

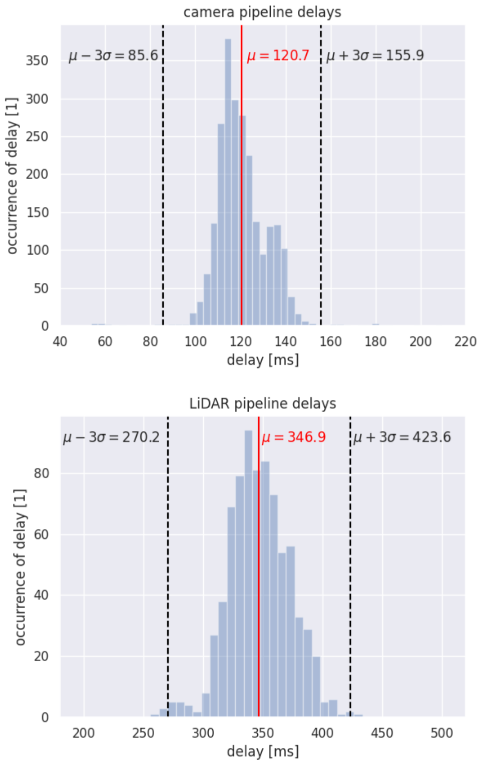


Fig. 10: Distribution of the end-to-end delays of our tracking pipelines (camera top, LiDAR bottom).

frequency of maximum $10Hz$ and transmits them to a V2X on-board unit in an automated vehicle nearby. Fig. 9 (iii) shows the received CPM trajectories that are based on the camera trajectories (see Fig. 9 (i)). There are two effects that can be noted during the experiment. First of all, it can be observed that uniform sub-sampling introduces artefacts that might be mitigated by implementing improved sub-sampling procedures as recommended in the CPM draft [7] or [33] or approaches like our previously used custom sub-sampling technique applied in [11] that uses derivation from a constant motion model to determine the frequency to sample objects in the object list. Secondly, after CPM transmission small discretization errors on localization information can be observed on transmitted points. This can be explained by the datatype encoding of the *DistanceValue* field in the CPMs. This field which encodes location information for objects, has a limited precision of at most $1cm$. Hence, all object locations are naturally discretized to multiples of $1cm$.

VII. CONCLUSIONS

In this work, we introduced *Infra2Go*, a flexible mobile research platform for data acquisition, testing and evaluation of highly autonomous driving systems that may be supported by intelligent infrastructure.

The Infra2Go platform is designed to provide flexibility when selecting test sites since it can be easily transported. It provides a hardware and software platform to perform reference data recording and testing of realtime algorithms for infrastructure based driving that can rely on cooperation between vehicles and infrastructure.

Our baseline methods for vulnerable road user tracking, that are based on two independent sensor streams, are a feasible approach for realtime trajectory estimation and have been demonstrated in a relevant real world scenario. First steps towards collaborative perception between infrastructure and an automated vehicle have been performed by evaluating V2X communication of perceived vulnerable road user trajectories as Cooperative Perception Messages between a Roadside Unit in our platform and an on-board V2X Unit in a vehicle.

The Infra2Go platform serves as a general purpose tool for testing, evaluation and validation of methods for connected automated driving in various aspects. This may include the evaluation of sensor fusion algorithms between infrastructure and vehicles or different sensor modalities like *thermal imaging cameras*, *dynamic vision sensors* or *Radar Detection and Ranging* for infrastructure sensing. Furthermore the device can be used as a standalone evaluation and validation platform for the validation in field tests and can be applied in test areas and proving grounds in the future.

ACKNOWLEDGMENT

This work was done within the project ”Digitales Testfeld BW für automatisiertes und vernetztes Fahren”, referred to as ”Testfeld Autonomes Fahren Baden-Württemberg”, funded by the Ministry of Transport Baden-Württemberg.

REFERENCES

[1] F. Naujoks *et al.*, “Effectiveness of advisory warnings based on cooperative perception,” *IET intelligent transport systems*, vol. 9, no. 6, pp. 606–617, 2015.

[2] M. Gabb *et al.*, “Infrastructure-supported perception and track-level fusion using edge computing,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 1739–1745.

[3] M. Shan *et al.*, “Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations,” *Sensors*, vol. 21, no. 1, p. 200, 2021.

[4] A. P. Tarko *et al.*, “Tscan: Stationary lidar for traffic and safety studies—object detection and tracking,” 2016.

[5] A. Carreras *et al.*, “Road infrastructure support levels for automated driving,” *25th ITS World Congress*, Sep. 2018.

[6] “Final draft ETSI EN 302 637-2 V1.3.1 (2014-09) Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service,” European Telecommunications Standards Institute (ETSI), Standard, Sep. 2014.

[7] “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2,” European Telecommunications Standards Institute (ETSI), Standard, Dec. 2019.

[8] “Intelligent Transport Systems (ITS); Vulnerable Road Users (VRU) awareness; Part 2: Functional Architecture and Requirements definition; Release 2,” European Telecommunications Standards Institute (ETSI), Standard, Dec. 2021.

[9] L. Liu *et al.*, “Equinox: A road-side edge computing experimental platform for cavs,” in *2020 International Conference on Connected and Autonomous Driving (MetroCAD)*, 2020, pp. 41–42.

[10] S. Khan *et al.*, “Towards collaborative perception for automated vehicles in heterogeneous traffic,” in *International Forum on Advanced Microsystems for Automotive Applications*, Springer, 2018, pp. 31–42.

[11] T. Fleck *et al.*, “Towards large scale urban traffic reference data: Smart infrastructure in the test area autonomous driving baden-württemberg,” in *International Conference on Intelligent Autonomous Systems*, Springer, 2018, pp. 964–982.

[12] T. Fleck *et al.*, "Robust tracking of reference trajectories for autonomous driving in intelligent roadside infrastructure," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1337–1342.

[13] D. Notz *et al.*, "Extraction and assessment of naturalistic human driving trajectories from infrastructure camera and radar sensors," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2020, pp. 455–462.

[14] S. Masi *et al.*, "Augmented perception with cooperative roadside vision systems for autonomous driving in complex scenarios," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 1140–1146.

[15] A. Krämmer *et al.*, "Providentia-a large scale sensing system for the assistance of autonomous vehicles," in *Robotics: Science and Systems (RSS), Workshop on Scene and Situation Understanding for Autonomous Driving*, 2019.

[16] N. Scheiner *et al.*, "Automated ground truth estimation of vulnerable road users in automotive radar data using gnss," in *2019 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, IEEE, 2019, pp. 1–5.

[17] L. Kloeker *et al.*, "Traffic detection using modular infrastructure sensors as a data basis for highly automated and connected driving," in *29. Aachen Colloquium-Sustainable Mobility*, vol. 29, 2020, pp. 1835–1844.

[18] M. Hoss, M. Scholtes, and L. Eckstein, "A review of testing object-based environment perception for safe automated driving," *Automotive Innovation*, pp. 1–28, 2022.

[19] C. Creß and A. C. Knoll, "Intelligent transportation systems with the use of external infrastructure: A literature survey," *arXiv preprint arXiv:2112.05615*, 2021.

[20] P. Schörner *et al.*, "Park my car! automated valet parking with different vehicle automation levels by v2x connected smart infrastructure," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 836–843.

[21] J. Weimer *et al.*, "Managed automated driving: A new way for safe and economic automation," in *27th ITS World Congress*, 2021.

[22] *SAPOS – Satellitenpositionierungsdienst der Deutschen Landesvermessung*, `https://sapos.de/`, Accessed: 2022-03-01.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[24] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[25] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020.

[26] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," en, *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006, Number: 11, ISSN: 1053-587X.

[27] K. Panta, D. E. Clark, and B.-N. Vo, "Data Association and Track Management for the Gaussian Mixture Probability Hypothesis Density Filter," en, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1003–1016, Jul. 2009, ISSN: 0018-9251.

[28] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5067–5073.

[29] F. Hasecke, L. Hahn, and A. Kummert, "Flic: Fast lidar image clustering," in *ICPRAM*, 2021.

[30] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3d laser scans for online operation," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[31] ——, "Efficient online segmentation for sparse 3d laser scans," *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, pp. 1–12, 2017.

[32] F. Poggenhans *et al.*, "Lanelet2: A high-definition map framework for the future of automated driving," in *Proc. IEEE Intell. Trans. Syst. Conf.*, Hawaii, USA, 2018.

[33] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Generation of cooperative perception messages for connected and automated vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, 16336–16341, 2020, ISSN: 1939-9359.