

Code #LikeABosch

Software development challenge

Your task for this weekend is not only to track objects around the vehicle and provide collision warning in the blind spots but also to be the innovator, the idea owner and show to the world the next generation of driver assistance.

Who we are?

Bosch is one of the market and technology leaders in the automotive industry, in nearly all areas connected to vehicles and transportation to our partners. We provide software, hardware, and system solutions in the field of vehicle safety, vehicle dynamics, and driver assistance. Through the combination of systems based on cameras, radar and ultrasonic sensors, electric power steering and active or passive safety systems we not only make driving safer and more comfortable today but we are also building the basis for the autonomous driving of tomorrow.

Problem description

Driver assistance systems are continuously being developed to support the driver's safety and convenience more and more. These systems consist of 4 main layers:

- Sensing layer: using camera and radar sensors
- Perception layer: detection and classification of the objects around the vehicle
- World model layer: object tracking and estimating the behaviour of our environment
- Function layer: based on all of the above make decisions how our vehicle shall behave, i.e., features

Research and developments are heavily and parallelly ongoing in all layers to provide more advanced safety and comfort to future vehicles, and of course to strengthen and develop the company's market position. Now we are inviting you to join these tasks in the 'World model layer' and the 'Function layer'.

For now, in most new vehicles a so-called 'blind spot detection' feature is available, which focuses on the blind spot of the driver (due to side mirror settings). However, we aim to gather sensor information in 360° around the vehicle, there are areas that are out of the sensors' field of view (FoV) – sensor blind spots.

The challenge

There are several parts to this challenge. It is not mandatory to complete all tasks, it's up to you which parts you target. You can also choose to skip some parts, but focus on the more precise implementation or presentation of others, or just have a more high-level solution for all parts. The details of the evaluation are shared at the end of this document in 'Judging criteria'. We encourage you to focus on those parts which are sparking the highest interest in you and give you the most joy and satisfaction during the next 2 days.

In this challenge the following sensors will be used:

- Front video camera
 - Maximum 10 objects provided at the same time
- Corner radars (1 radar at each corner)
 - Maximum 10 objects provided at the same time

In **The first part** you can build up an object tracking system all around the vehicle and show, how well your solution can track the objects even in case they are in sensor blind spots – out of the sensors' FoV.

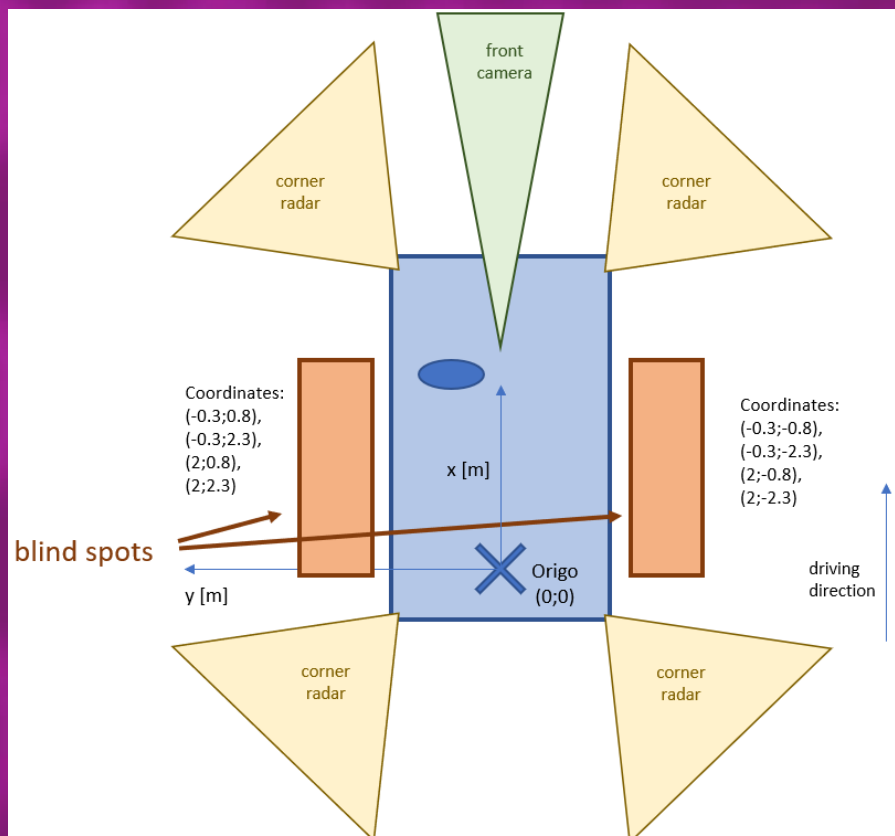
Our expectation is that you can present (preferably in a simple visualization) the object positions sensed by the sensors, and your own estimated object around the vehicle.

- Based on sensor inputs create your own estimated objects around the vehicle
- Track these objects over time, so update the estimated object information in the next timestamp

It's possible that one object is sensed by more sensors, not only one. In this case it's up to you how you recognize if two sensors detect the same real object. Here're some ideas:

- Just take the absolute distance of the detected objects. If it's below a certain limit, they can be considered as one
- Use a Kalman-filter (a more advanced solution)

You will receive a database (in the format of .csv) from real vehicle measurements, which will contain object information (position, velocity, acceleration) from all sensors around the vehicle (front camera and 4 corner radars).



The second part is to create and implement the ‘sensor blind spot’ functionality. The blind spot is defined as an area by the coordinates in the vehicle’s own coordinate system. When the object (prediction of the object’s location) enters the blind spot in the designated area (orange in Figure 1), and the driver’s intention is to turn (turning indicator is ON), show an alert (either acoustic or optic). This blind spot is not fully covered by the sensors, therefore to react on all objects in the blind spot, some object-tracking is needed.

For some extra fun, points and acknowledgement, the advanced version of this part is to give warnings only on objects that are classified as vehicles. Note, that the object classification information (whether an object is a pedestrian, cyclist, vehicle or unknown) is only available at the video sensor. So here your object tracking needs to also bring along the information gathered by the front video (the object classification) even when the object can be seen by only the corner sensors, not the front video.

In the **third part** you’ll need your creativity and innovative thinking. Your task is to come up with ideas for new features and functionalities, that can be realized with the above sensor set and object tracking solution – preferably functionalities, that are today not yet on the market.

We are curious about your ideas, their technical details, your business plan – why your idea is good and the market needs it, how you present it to us, and of course if a prototype implementation is created, the presentation of the functionality.

What we will provide

Three kinds of datasets will be used during the Hackathon.

- Development dataset
- High accuracy dataset
- Validation dataset

Development dataset: You will be provided a dataset which you can use for your development.

High accuracy dataset: Additionally, you’ll get a smaller dataset, which includes also dGPS (high accuracy position info) data about a vehicle around ‘our’ vehicle. Feel free to use this dataset as well, for validating and improving your object tracking implementation.

Validation dataset: This will be available only for the mentors and the jury, and for you after the finalization of the tasks. Checking your solutions on this dataset makes sure, that you don’t overoptimize your implementation to your development dataset ☺

The datasets are in .csv format, and contains the following info:

- Timestamp (synchronized for all sensors)
- Front video: object position, velocity, acceleration, classification
- 4 corner radars: object position, velocity, acceleration

- Ego vehicle information ('our' vehicle is called ego), such as velocity, acceleration, yaw rate, direction indicator status, and some more

Implementation and technology

As we are working on embedded systems at Bosch, we mostly use C++, but you are free to use any kind of technologies and languages to solve the challenges above.

Judging criteria

First part of the task: object tracking

- Accuracy of the object tracking even in blind spots (using the validation dataset)
- Presentation (method, visualization, presentation style)

Second part of the task: feature development

- How precisely your solution gives the warnings (e.g., only when the objects are in the designated area, etc....)
- If the warning comes only on pedestrians and cyclists or on all objects
- Presentation (method, visualization, presentation style)

Third part of the task: new features

- Level of elaboration of the new idea(s)
- Innovativeness
- impact/value, business case
- sustainability
- feasibility
- prototype implementation
- presentation