Code #LikeABosch

Software development challenge

Your task for the upcoming 24 hours will include implementing a solution for vehicle collision avoidance as well thinking like an innovator and delivering the next generation of driver assistance systems. Join us on this exciting challenge!

Who we are?

Bosch is one of the market and technology leaders in the automotive industry, in nearly all fields of 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, radars, 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.

Background

Driver assistance systems are continuously being developed to support the driver and other road participants more and more. These systems consist of 4 main layers.

- Sensing Layer: using camera, radar, and sometimes lidar sensors
- Perception layer: detection and classification of the objects around the vehicle
- Situation layer: object tracking and estimating the behaviour of the car's environment.
- Function layer: based on all the above make decision how our vehicle shall behave

Research and development 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 'Perception layer', 'Situation layer' and the 'Function layer'.

For now, most of the vehicles (from 07.2024 all of them) have to have a function which is called Automatic Emergency Brake (AEB). A car, which has AEB can avoid, or mitigate collision automatically, without any driver intervention. There are several scenarios, which are handled by AEB. This can work with one sensor (1 radar, or 1 video camera), or in a fusion mode, where the camera and the radar provide fused objects (both of them see the object), which will be processed by Perception. That layer identifies the object, what it is (vehicle, pedestrian (adult, child), motorcyclist). After that,it decides if the target is relevant for the system, or not. After that, the Situation layer will process this information, and evaluate, what is the scenario (turning, longitudinal, crossing). If the identification happened, they decide if the Function layer should react for these objects in the evaluated scenario, or not. If the answer is yes, then the Function layer will start the reaction, and try to avoid, or at least mitigate the collision.

Challenge

There are 3 parts to the challenge. It is not mandatory to complete all the tasks, it's up to you which part, or parts you target, but please be aware, if you don't do the 1st part, you won't be capable of doing the second and the third part of the challenge. You can skip the second, and/or the third part, the main goal is to be as precise as you can in the implementation and in the presentation as well. The challenge details will be shared at the end of the document in 'Judging criteria'. Don't forget, you are here to have

fun, and show your expert skills! We are pretty sure, the challenge will be really a challenge, but at the same time, it will be a lot of fun.

In this challenge the following sensor will be used: **Front video camera**

In the first part you have to filter out all the unnecessary target objects, and find with the visualization the correct scenario (the scenarios are listed in the second part of the challenge). You have to measure the longitudinal and lateral distance of the relevant object from the vehicle, and the speed of it. From this information you have to create the environment of the vehicle, how the object and vehicle move, and define the scenario.

Our expectation is that you can present (visualize) the environment of the vehicle, the relevant object, the vehicle's and target object's speed. So all in all, all the motions and the full environment and the scenario should be visualized.

The visualization can be anything, what you know, and capable of that. For example (python script, .NET, UnrealEngine, Javascript etc..)

- Based on the available data create a visualization of how the vehicle and the relevant object move, and the full environment

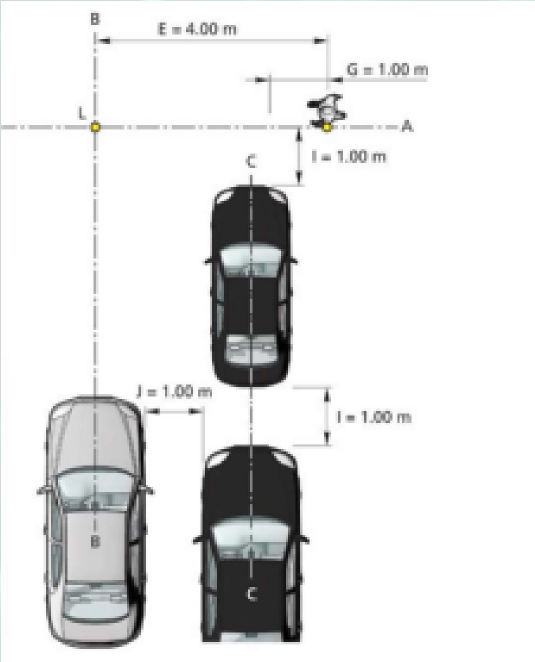
- Filter out all the unnecessary target objects, calculate the speed of the vehicle timestamp to timestamp, and as well the target object's speed, the longitudinal and lateral distance from the vehicle.

You will get a database (in the format of .csv) from a real vehicle measurement, which will contain all the relevant information (objects, speeds, distances).

The second part is to decide about the first party's information, and what kind of scenario happens based on your visualization. Here are the chooseable scenarios:

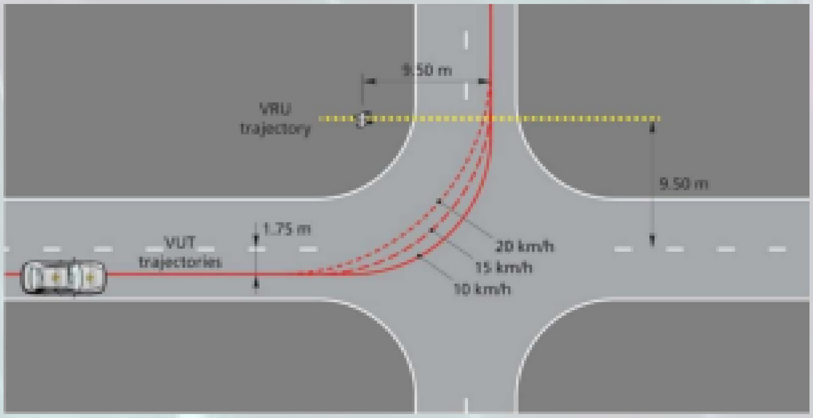
- CPNCO – Car to Pedestrian Nearside Child Obstructed

A collision in which a vehicle travels forwards towards a child pedestrian crossing its path running from behind and obstruction from the nearside and the frontal structure of the vehicle strikes the pedestrian at 50% of the vehicle's width when no braking action is applied.



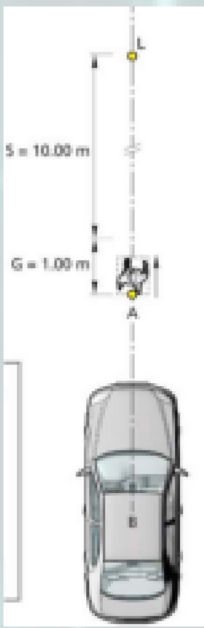
- CPTA – Car to Pedestrian Turn Adult

A collision in which a vehicle turns towards an adult pedestrian crossing its path, walking across a junction (in either the same and opposite direction as the VUT, before the VUT made the turn) and the frontal structure of the vehicle strikes the pedestrian at 50% of the vehicle's width when no braking action is applied.



- CPLA – Car to Pedestrian Longitudinal Adult

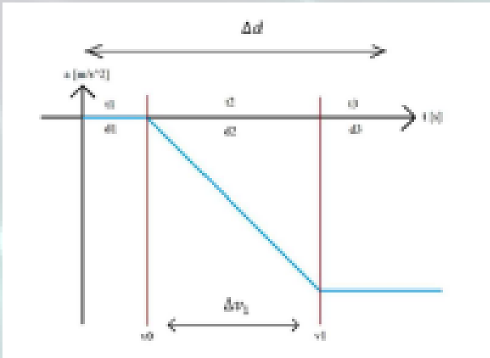
A collision in which a vehicle travels forwards towards an adult pedestrian walking in the same direction in front of the vehicle where the vehicle strikes the pedestrian at 50% of the vehicle's width when no braking action is applied.



If you identified the scenario you can move to the third part of the challenge.

The third and final part is to implement a solution, which can avoid the collision. The solution is totally up to you, how you would like to do it. Just some hints:

- Signal to the target object with the horn
- Flash the front headlights to the object
- Flashing/sound alarm of the pedestrian crossing light
- Calculate the brake distance, and how much deceleration needed to avoid the collision. For that, here are some help:



BrakeDistance Calculation

$\Delta d = d_1 + d_2 + d_3$

$d_1 = v_{ego} * t_{lat} + \frac{a_{ego}}{2} * t_{lat}^2$

$d_2 = \frac{maxJerk}{6} * t_2^3 + \frac{a_{ego}}{2} * t_2^2 + v_{ego} * t_2$

$d_3 = \frac{a_{max}}{2} * t_3^2 + v_1 * t_3$

$t_2 = \frac{a_{max} - a_{ego}}{maxJerk}$

$t_3 = \frac{\Delta v_2}{a_{max}}$

$\Delta v_2 = \quad - v_1$

$v_1 = v_0 + \ \Delta v_1$

$\Delta v_1 = \frac{maxJerk}{2} * t_2^2 + a_{ego} * t_2$

Hint: For maxJerk, use $-30 m/s^3$, aEgo is the acceleration/deceleration of the vehicle, aMax is the maximum deceleration. Use $-9\, m/s^2$. vEgo is the speed of the vehicle.

Use your creativity! The main goal of this task is to avoid collisions! But it can be accepted, if you provide a theoretical solution.

What we will provide

Two kinds of datasets will be used during the Hackathon:

- ★ **Development dataset**
- ★ **Validation dataset**

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

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

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

- timestamps
- objects, object positions, velocities

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 challenge above.

Judging criteria

First part of the challenge: Object filtering and environment/situation build-up

- You can present the solution visually
- Only the relevant object and the vehicle are on the visualized picture/video/gif

Second part of the challenge: Situation decision

- You were able to choose which scenario happened about the data

Third part of the challenge: Avoid or mitigate the collision

- You were able to implement a solution, which can avoid or mitigate the collision with the target object
- Not necessary to implement a visual solution, it's enough to be able to share your thoughts and prove the idea

Innovativeness

Feasibility

- We would like to see a walkthrough step by step as a presentation