SW-DEVELOPMENT FOR AUTONOMOUS DRIVING FUNCTIONS @ AVL SFR

05.07.2017
AGENDA

1. Autonomous Driving Architecture
2. Camera perception using CNN
3. Trajectory design Highway Chauffeur
4. Testing Environment
AVL ADAS & Autonomous Driving Functions

Inputs
- (Stereo) Camera
- Radar
- LIDAR
- Maps / Navi

Perception
- Vehicle State
- Sensor-Fusion
- Occupancy-Grid
- Localisation

Outputs
- Steering
- Braking
- Acc.-Pedal
- HMI

Arbitration / Decision Making
- FCC/LKA / LCA
- Maneuver Planning
- Decision Making / Evaluation

Emergency States
- Trajectory Panning
- Level 4 & 5
- Level 3
- Up to Level 3

Perception
- Perception
- Sensor-Fusion
- Occupancy-Grid
- Localisation

Arbitration / Decision Making
- FCC/LKA / LCA
- Maneuver Planning
- Decision Making / Evaluation

Emergency States
- Trajectory Panning
- Level 4 & 5
- Level 3
- Up to Level 3
MACHINE LEARNING / NEURAL NETWORKS = NON-DETERMINISTIC FUNCTIONS

Choose NN

- Convolutional Layers: number of Kernels / size of Kernel
- Pooling Layer: max / sum
- Non-linearities: Relu / Elu / sigmoid / tanh

Input-DATA

DATA balanced

yes

Train

Results

no

DATA Augmentation
# SPEED COMPARISON STATE-OF-THE-ART OBJECT DETECTOR USING DEEP LEARNING

<table>
<thead>
<tr>
<th>Model</th>
<th>Car</th>
<th>#</th>
<th>Pedestrian</th>
<th>Cyclist</th>
<th>#</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Easy</td>
<td>Moderate</td>
<td>Hard</td>
<td>Easy</td>
<td>Moderate</td>
<td>Hard</td>
</tr>
<tr>
<td>YOLO</td>
<td>35,86%</td>
<td>49,47%</td>
<td>29,74%</td>
<td>24,35%</td>
<td>25,63%</td>
<td>17,50%</td>
</tr>
<tr>
<td>YOLOv2</td>
<td>69,01%</td>
<td>86,40%</td>
<td>59,57%</td>
<td>43,33%</td>
<td>53,02%</td>
<td>35,41%</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>79,11%</td>
<td>87,90%</td>
<td>70,19%</td>
<td>65,91%</td>
<td>78,35%</td>
<td>61,19%</td>
</tr>
<tr>
<td>MM-MRFC</td>
<td>88,20%</td>
<td>90,93%</td>
<td>78,02%</td>
<td>69,96%</td>
<td>82,37%</td>
<td>64,76%</td>
</tr>
<tr>
<td>WRInception</td>
<td>87,62%</td>
<td>88,98%</td>
<td>77,52%</td>
<td>68,76%</td>
<td>79,98%</td>
<td>63,48%</td>
</tr>
<tr>
<td>FD2</td>
<td>74,68%</td>
<td>87,68%</td>
<td>65,70%</td>
<td>56,68%</td>
<td>71,09%</td>
<td>51,65%</td>
</tr>
</tbody>
</table>

Benchmark results based on KITTI object detection dataset (note: different HW used!), no SSD300 or SSD500 benchmarks provided on this dataset.

1) Autonomous submission, no code/paper provided, used additional flow (2 adjacent images) and point cloud information
2) Autonomous submission, no code/paper provided
3) Autonomous submission, no code/paper provided
SSD 300 DETECTION EXAMPLE
VALIDATION / IMPROVEMENT OF NON DETERMINISTIC FUNCTIONS

- Big Data Processing (Backend-Server)
- Neural Network Training, Validation and Testing
- Virtual Validation of overall system (5 Mio. Driving scenarios)
- Software update by Backend-Server

Complete Vehicle Fleet provides Sensordata

Software over the air update
AVL has chosen Quintic polynomial approach for overtake functions, it ensures/ supports:

- Highly modularity – also stand alone
- Function is easy to execute and cancel

**Maneuver split:**

- Host vehicle moves to faster lane
- Host vehicle continues in the faster lane
- Host vehicle returns to the slower lane

**Trajectory Planning – Level 3**

Solve 5\textsuperscript{th} degree polynomial and applying boundary conditions, the coordinates of the trajectories are obtained by

\[
x(t) = Vt + (Vt - D) \left[-10t^3 + 15t^4 - 6t^5\right]
\]

\[
y(t) = W + W \left[-10t^3 + 15t^4 - 6t^5\right]
\]

Where T and D are solved using constrained optimization problem

Optimum distance for maneuver \( D^* = 2.4V\sqrt{\frac{W}{A}} \)

Optimum time for maneuver \( T^* = 3\sqrt{\frac{W^{3/2}\sqrt{A}}{V^2}} + 2.4\sqrt{\frac{W}{\sqrt{A}}} \)

TRAJECOTORY PLANNING

Maneuver completion

Maneuver abortion due to fast vehicle approaching on acceleration lane.
TEST ENVIRONMENT

Combining AVL VSM with PreScan

Vehicle model – Driver – 3 d Visualization
AVL Inhouse tool / platform
IODP – Model.connect

Sensors - Environment – Traffic - Infrastructure
Leading sensor models
DIL SETUP - ENVIRONMENT

Basic Simulator Setup

Simulator Seat
(Control Area / User Inputs)

Vehicle Control
- Steering
- Throttle Control
- Brake Control
- Switch
- ...
- ...

HMI Control
- Volume Button
- Speed Control
- ...
- ...
- ...

Actuator Control PC

CAN Transceiver

Simulator Server

CAN Interface

GPIO Controls

Ethernet Gateway

AVL Simulation Environment

AVL Simulation

Display

HMI

Ethernet Transceiver Interface

UDP

Model in Loop
- Controls
- PreScan
- VSM
Highway Chauffer End of 2017:

- Highly automated Driving (SAE Level 3 [SAE, 2014])
- Application on highways and highway-like streets
- Velocity range between 0 and 130 kph
- Automated lane change and take over maneuver
- Stop & Go
- Adequate processing of emergency situations

NVIDIA PX2:

- Execution of „Convolutional Neural Networks“ for Camera-Object detection and classification
- Sensorfusion of Radar-Object lists with Camera-Object lists

dSpace MicroAutobox 2:

- Integration of LKA / LCA / FCC
- Execution of longitudinal and lateral control
SUMMARY, WHY AVL?

- Proven solutions up to series production, successfully implemented at various car manufactures
- Development & testing tools to achieve best in class driving quality and perceived safety
- Innovative predictive energy efficiency concepts with significant fuel savings
- Global AVL engineering network enables efficient projects on/off-site at customers depending on demand
A typical list \((A,B,C,D\ldots)\) consists of:

1. **ID** (ID of the object)
2. **TrackState** (valid or not)
3. **x** (longitudinal position)
4. **y** (lateral position)
5. **vx** (longitudinal velocity)
6. **vy** (lateral velocity)
7. **ax** (longitudinal acceleration)
8. **ay** (lateral acceleration)
9. **P** (Error Covariance Matrix)
10. **UpdateTime**

**Kinematic Models available**

1. Constant \(v\)
2. Constant \(a\)
3. Constant turnaround
APPROVED TOOL CHAIN FOR “CONVENTIONAL” (CONTROLS-) FUNCTIONS
LONGITUDINAL / LATERAL CONTROL
LONGITUDINAL / LATERAL CONTROL

Unscented KF

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\psi}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\dot{v}_{LV} \\
\dot{\psi}_{LV}
\end{bmatrix}
\]

\[
\begin{bmatrix}
v_E \\
\dot{\psi}_E
\end{bmatrix}
\]

IMF-Fusion

V2X

\[
\begin{bmatrix}
a_{LV} \\
a_{SP}
\end{bmatrix}
\]

Feedforward

+ Disturbance

Feedback

Distance Setpoint

[\text{V2X}] 

CAN

[\text{RADAR}]

Lead Vehicle (LV)

Ego Vehicle (E)
CONVENTIONAL COMPUTER VISION FOR DISTANCE ESTIMATION → GENERATION OF OBJECT LIST

• For calculating the position of the detected and classified objects triangulation is necessary.

\[ Z = \frac{b \cdot f}{x_R - x_T} \]

SIMULATION

- Curved road ($D = 300$ m)
- Blocked field of view
  - Following car 30 m
  - Several other vehicles
- ego car velocity: **100 km/h**
- overtaking car velocity: **200 km/h**
  → car detected in 90 m range