23322: Open Fusion Platform for Automated Driving Cars Based on Nvidia DPX2
Agenda

- Motivation
- Overview of the *Open Fusion Platform* (OFP) Project
- Functional Architecture
- Interface Specification
- Joint Semantic Segmentation and Detection
- Intention Prediction, Risk Assessment and Decision Making for Vehicle Guidance
- Conclusion and Outlook
Motivation

- Fully automated driving prototypes available but no serial production
  - High development cost on sensors, hardware and algorithm
- Advanced driver assistance systems cover only predefined situations
- Integration issues due to heterogeneous sensors and interfaces
- Actual standardization initiatives for automated driving
  - OpenDRIVE: Format specification for road networks and infrastructure
  - OpenSCENARIO: Description of dynamic contents in driving simulation applications
  - Adaptive AUTOSAR: New AUTOSAR Platform for complex fusion systems
  - EB Robinos
    - Architecture specification in environmental fusion models
    - Software for development and embedded prototyping
    - SW Modules inside environmental model and situation analysis
Overview of OFP

Project Objective:

- Create a near series fusion platform with open interfaces, that allows a cost efficient implementation of highly and fully automated driving functions.

Sensor Configuration

Project Timeframe: Jan 2016 – Dec 2018

Use Case:

“An e-car autonomously parks and positions itself directly on top of a parking space with a wireless charging plate. When car is charged, it drives itself to another parking space without a charging plate.”
Car Platforms used in OFP

- e.GO life (Electric)
- Passat GTE (Plug-in Hybrid)
- AUDI A6
Functional Architecture

Nvidia Drive PX2

Sensors
- Camera
- RADAR
- SV Camera
- Ultra Sonic

Perception
- Object Detection
- Object Classification
- Object Tracking
- Semantic segmentation
- Localization
- Ego Motion
- Intention Recognition
- Park Marking Detection

Fusion
- Semantic Scene Interpretation
- Map Fusion
- Free Space Detection
- Parking Space Detection
- Object Fusion
- Ego Fusion
- Consistency Checker

Fusion Framework
- Single-Sensor Data Processing
- Multi-Sensor Data Fusion & Interpretation

Supervision

Application
- Static Environment
- Dynamic Environment
- Vehicle State

Environment Model
- Map Data
- Ego Fusion

Actuators
- EPS
- Brake
- Motor
- HMI

Control Algorithms & Output Interfaces
- Scene Prediction
- Navigation
- Maneuver Planning
- Trajectory Planning
- Function Arbitration
- Function specific Interpretation
- Analysis & Action Planning
- State Machine
- Safety Manager
- Service Manager
- Libraries

Vehicle Data
- GNSS
- Car2X
- Map Data
Interface Specification

- Hardware Interface
  - Use available standards as CAN, LVDS, etc.

- Basic Software
  - Elektrobit AdaptiveCore: Adaptive AUTOSAR implementation from Elektrobit

- Data- and timing-driven communication

- Definition of generic data types. E.g. object, ego pose and motion, image, etc.

- Software interface
  - Description of inputs and outputs using module manifest
  - Specification of a module manifest template
  - Layer specific module manifest implements the manifest template

- Interface specification available soon on the project homepage [http://www.ofp-projekt.de/ofp-project/de/Oeffentliche-Dokumente-305.html](http://www.ofp-projekt.de/ofp-project/de/Oeffentliche-Dokumente-305.html)
Joint Semantic Segmentation and Object Detection

Overview

Sensors
- Camera
- RADAR
- SV Camera
- Ultra Sonic
- Vehicle Data
- GNSS
- Car2X
- Map Data

Perception
- Object Detection
- Object Classification
- Object Tracking
- Semantic Segmentation
- Localization
- Ego Motion
- Intention Recognition
- Park Marking Detection

Fusion
- Semantic Scene Interpretation
- Map Fusion
- Free Space Detection
- Parking Space Detection
- Object Fusion
- Ego Fusion
- Consistency Checker

Fusion Framework
- Single-Sensor Data Processing
- Multi-Sensor Data Fusion & Interpretation

Application
- Static Environment
  - Map Data
- Dynamic Environment
- Vehicle State
  - Driver State

Actuators
- EPS
- Brake
- Motor
- HMI
- Safety Systems

Environment Model

Supervision
- State Machine
- Safety Manager
- Service Manager

Control
- Algorithms & Output
- Interfaces

Libraries
- Calibration
- Synchronization
Joint Semantic Segmentation and Object Detection Model

- Sharing of features between Faster R-CNN [1] and FCN [2]
- Extension of Faster R-CNN ROI Poling with FCN score map
- Advantages
  - Reduce feed forward computation time and memory consumption
  - Improve detection while keeping the segmentation unchanged
Joint Semantic Segmentation and Object Detection
Training and Evaluation on Daimler Cityscapes dataset

- Training and evaluation on Daimler Cityscapes dataset [3] using GTX 1080ti GPU
- Evaluation results: Intersection Of Union (IOU) for segmentation and mean Average Precision (mAP) for detection

<table>
<thead>
<tr>
<th>FCN Model</th>
<th>Average IOU</th>
<th>FR-CNN Model</th>
<th>AVG mAP</th>
<th>Overall mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>0.579</td>
<td>Single</td>
<td>0.32</td>
<td>0.327</td>
</tr>
<tr>
<td>Joint Model</td>
<td>0.572</td>
<td>Joint Model</td>
<td>0.325</td>
<td>0.358</td>
</tr>
</tbody>
</table>

- Complexity on GTX 1080ti, image size: 2048x1024
  - The joint-Model uses 33% less memory and is 1.3x slower than both single models running in parallel

<table>
<thead>
<tr>
<th>Model</th>
<th>#Params</th>
<th>Runtime</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single FCN</td>
<td>134.5M</td>
<td>320ms</td>
<td>5.63GB</td>
</tr>
<tr>
<td>Single Faster R-CNN</td>
<td>136.9M</td>
<td>250ms</td>
<td>4.47GB</td>
</tr>
<tr>
<td>Joint-Model</td>
<td>256.9M</td>
<td>430ms</td>
<td>6.71GB</td>
</tr>
</tbody>
</table>
Joint Semantic Segmentation and Object Detection
Fine Tuning on OFP Surround View Camera Data

- Fine tuning of the model trained on Daimler Cityscapes with OFP surround view camera data

- Complexity on GTX 1080ti, image size: 1024x440
  - The joint-Model uses 23% less memory and is 1.08x slower than both single models running in parallel

<table>
<thead>
<tr>
<th>Model</th>
<th>Runtime</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single FCN</td>
<td>180ms</td>
<td>2.22GB</td>
</tr>
<tr>
<td>Single Faster R-CNN</td>
<td>112ms</td>
<td>1.98GB</td>
</tr>
<tr>
<td>Joint-Model</td>
<td>195ms</td>
<td>3.22GB</td>
</tr>
</tbody>
</table>

- [Video](#)
An Abstract Functional Architecture for Automated Driving

For an automated vehicle guidance, the model and understanding about the current scene and its context are required.
OFP-Platform Provides us with Required Information about the Current Scene with different APIs
Intention Detection and Prediction of Road Participants by Using Probabilistic Approaches such as Bayesian Network

Approach:
Dynamic data: $v_{lat}$, $v_{long}$, $a_{lat}$, $a_{long}$, ...  
Context data: $t_{time-to-turn}$, $t_{time-to-vehicle}$, $t_{time-to-line}$

Conversion of Bayesian Network to Junction Tree for better calculation performance

Calculation of probability of each Maneuver based using Junction Tree Algorithm

Maneuver Catalog:
- Follow Vehicle
- Follow Road
- Turn
- Lane Change
- Target Brake
- Trash

Some Results:
Simulation Results
Risk Assessment of the Scene by Calculating the Predicted Trajectory and Collision Probabilities

Motion Model in Road Coord. Sys. | Two Step Collision Check | Calculating the Collision Probabilities | Sampling Using Monte Carlo

Follow Vehicle | Follow Road | Turn | Lane Change | Target Brake | Trash

Gap Keeping Model | Constant Acceleration Model | Constant Radius Model: Constant Velocity Model + Constant Acceleration Model | Half Sinus Model: Constant Velocity Model | Constant Acceleration Model (Considering the Distance to Obstacle) | Constant Acceleration Model + Constant Yaw Rate Model

\[ b' = \text{Modell}(s,a) \]
Decision Making under Uncertainties by Using POMDP-Approach

POMDP
Partially Observable Markov Decision Process:

\[ S = \{s\} = \{x, y, v_x, v_y\} \] (States)

\[ A = \{a\} = \{Acc, CV, Dec\} \] (Actions)

\[ T = P(s'|s, a) \] (Transition)

\[ R = R(b, a) \] (Rewards)

\[ Z = \{z\} \] (Measurements)

\[ O = P(z'|s', a) \] (Observation)

\[ \gamma \in ]0; 1[ \] (Discount factor)

**Action Set A**

- \(a_0\): Acceleration
- \(a_1\): Deceleration
- \(a_2\): Constant Velocity

**State S**

- **Ego Vehicle**
  - Position \((x, y)\)
  - Velocity \((V_x, V_y)\)

- **Road Participants**
  - Position \((x, y)\)
  - Velocity \((V_x, V_y)\)

**Intention Model**

Evolution of the Scene in each frame

Reduce search space by eliminating branches with low rewards or too many action changes

Simulation Results
Conclusion and Outlook

- Open fusion platform based on Nvidia DPX2
- Functional architecture as a layer model
- Interface specification based on available standards and generic data types
- Joint learning of semantic segmentation and detection improves the detection
- Automated vehicle guidance using environment model from OFP platform

Next steps
- Integration into the DPX2 and test with live data
- Fine tuning of the models
- Validation of the algorithms in real scenarios
Thank you for your Attention!

Paulin Pekezou Fouopi, DLR, paulin.pekezoufouopi@dlr.de
Mohsen Sefati, RWTH Aachen, M.Sefati@pem.rwth-aachen.de
Dr. Michael Schilling, Hella, Michael.Schilling@hella.com

www.ofp-projekt.de

Open Fusion Platform

SPONSORED BY THE

Federal Ministry of Education and Research
