ROVISlaboratory

Elektrobit

Vision Dynamics for controlling autonomous vehicles

Prof. Sorin Grigorescu

Elektrobit Automotive Romania
Transilvania University of Brasov, Romania

World Usability Day
November 12, 2020

About Elektrobit (EB)

Technical competencies

EB's technical core competencies are development of automotive-grade (software) products and engineering services.

Global presence

Development and business offices in Austria, China, Finland, France, Germany, India, Israel, Japan, Romania, South Korea, and USA.

Employees

More than 2600 employees* worldwide.
Spans three continents and eleven countries.

Continental AG

Wholly owned, independent subsidiary of Continental AG.

Consistent growth

In 2018: +35 %

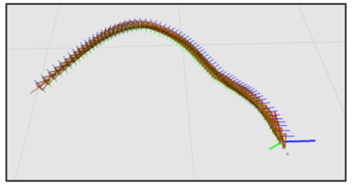
100+ million

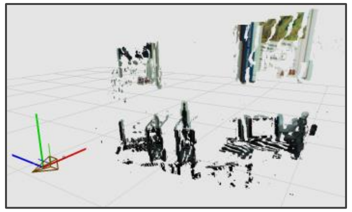
Over 100 million vehicles on the road and 1 billion embedded devices.

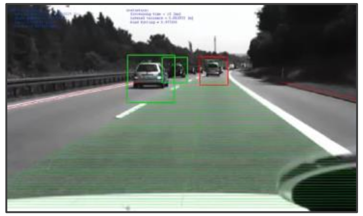
*December 2018, incl. Argus, excl. e.solutions.

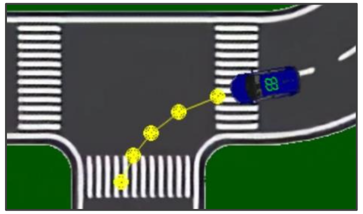
Scientific Achievements: Visual Control of Robots

- **From visual perception to motion planning and control**
- **Applications**: Self-driving cars (Elektrobit Automotive), FRIEND rehabilitation robot (University Bremen), PR2 service robot (Willow Garage, Google)









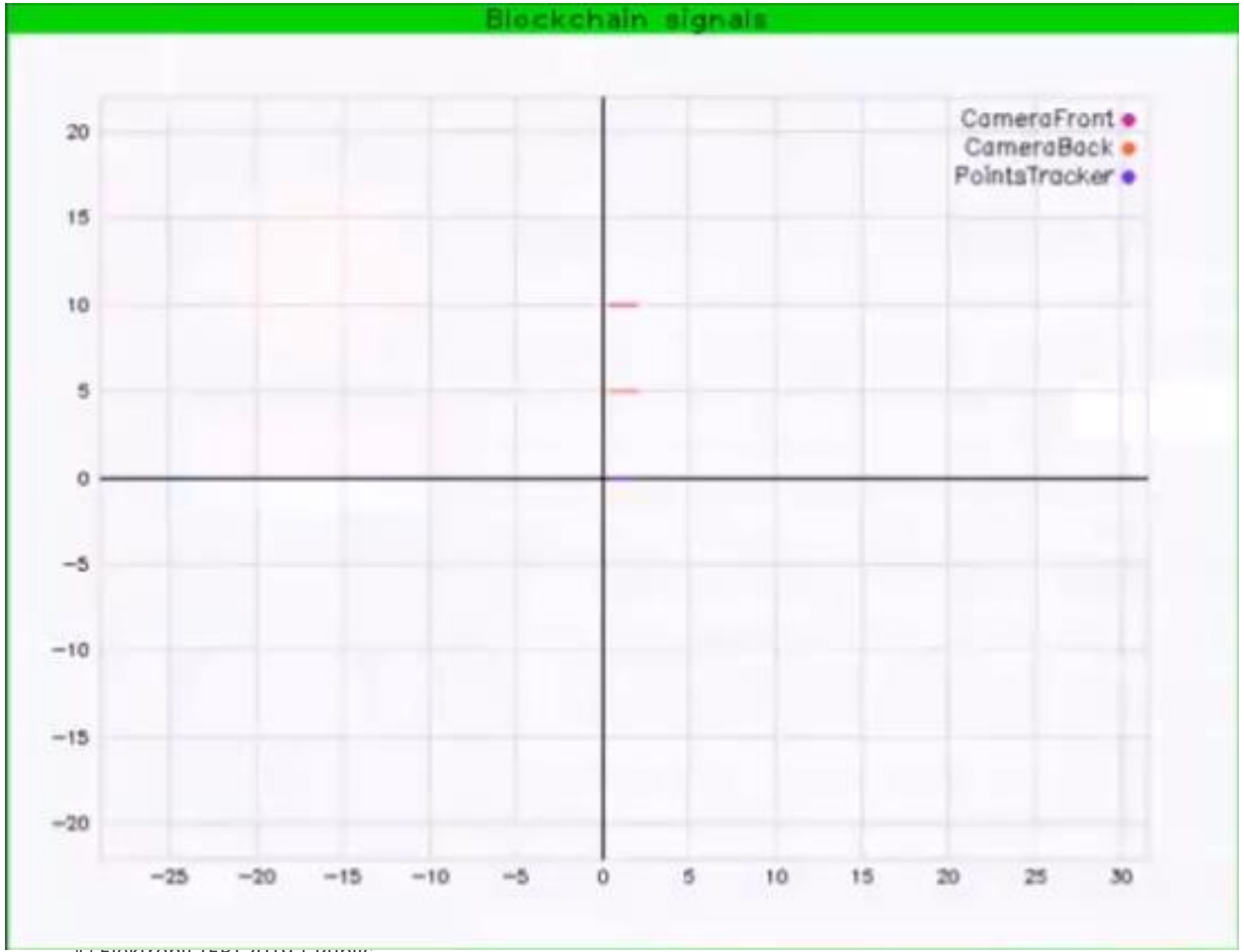
Visual Data Acquisition

- Stereo camera
- Time-of-Flight
- Structured light (MS Kinect)
- LiDAR

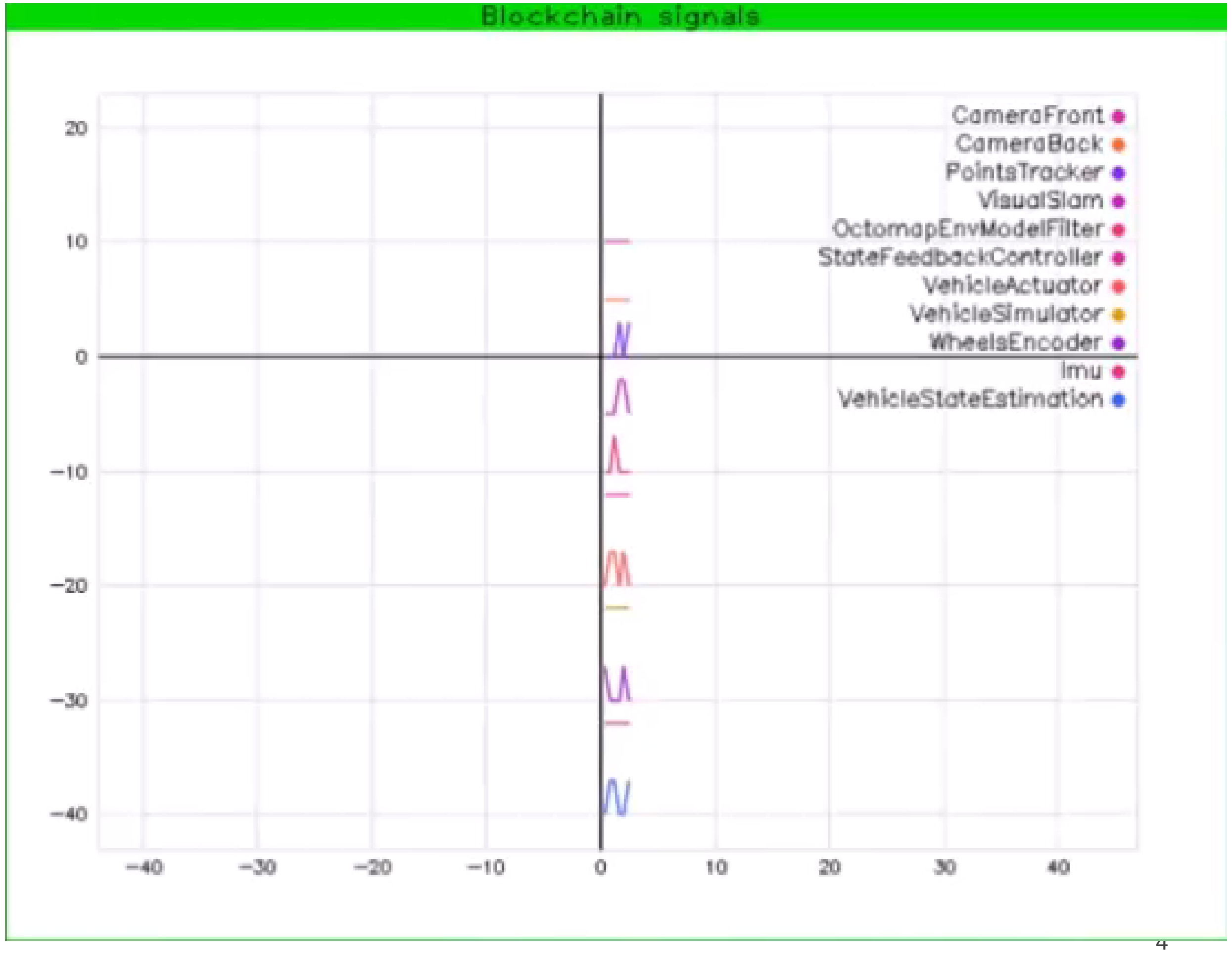
→ **Pose Estimation** → **3D Reconstruction and Object Tracking** → **Scene Understanding** → **Motion Planning and Control**

Computer Vision vs. Visual Robotic Control

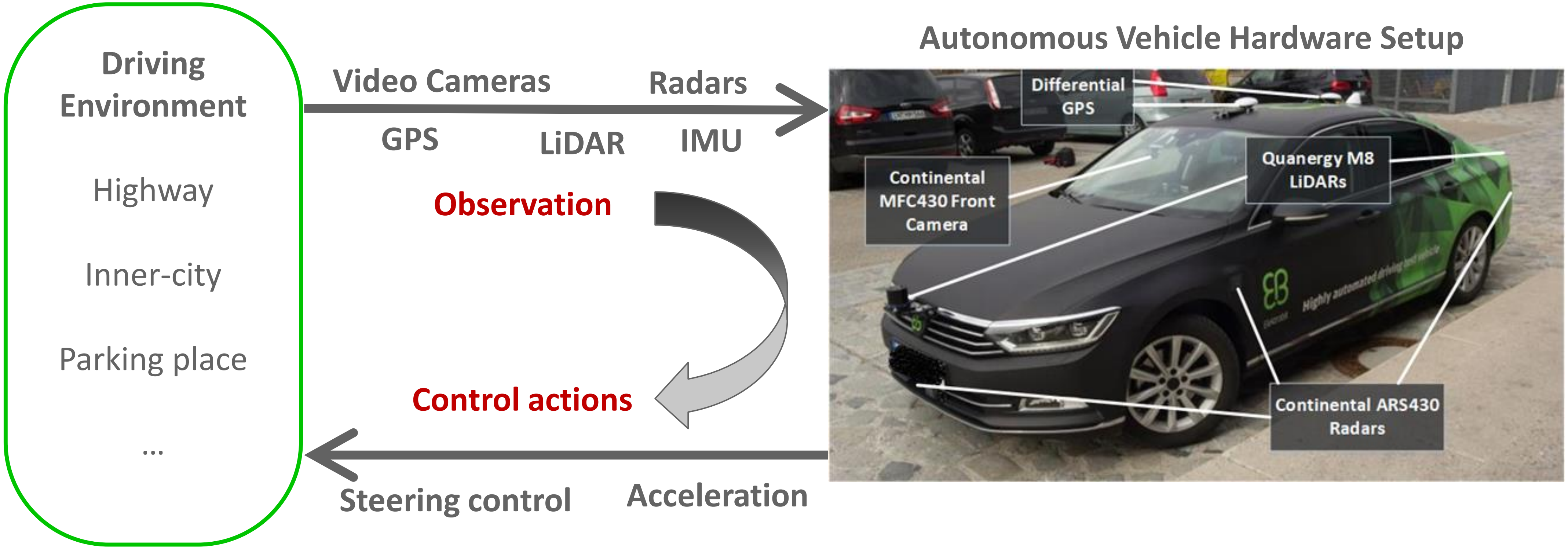
Computer Vision Task



Visual Robotic Control Task



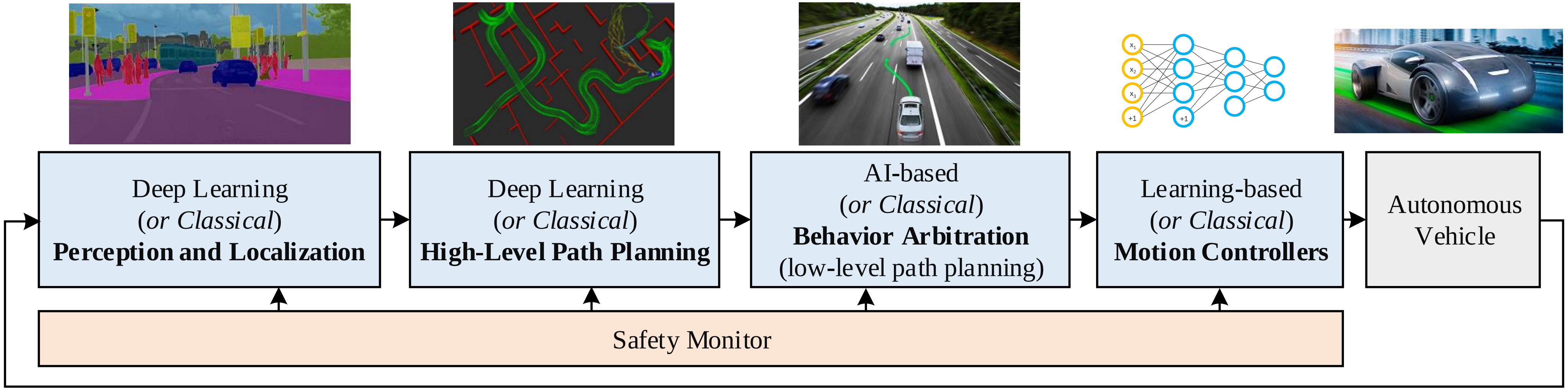
Robotic Systems as Intelligent Agents



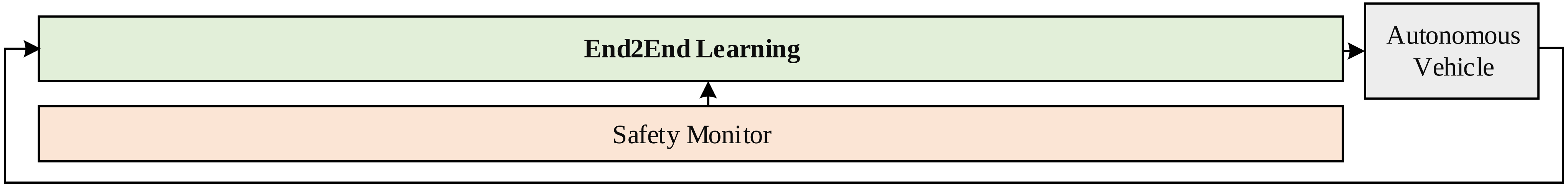
S.M. Grigorescu, M. Glaab and J. Schlosser, “KI für Selbstfahrende Autos”, *EE Faszination Elektronic*, 2017.

Autonomous Vehicle Control Pipeline

Perception-Plan-Action processing pipeline



End2End learning for direct sensors-to-actuation mapping



Grigorescu et. all, "A Survey of Deep Learning Techniques for Autonomous Driving", *Journal of Field Robotics*, 2019.

Better Learning-based Modules through AI

Learning-based modules

Non-learning modules

Artificial intelligence (AI) is intelligence exhibited by *machines*.

- “Artificial Intelligence" describes a machine that **mimics human cognitive functions**, such as learning and problem solving
- Currently dominated by **Machine Learning** and **Deep Learning** (large scale statistical learning systems)

Higher human brain functions simulation

Abstraction: deal with ideas

General programming language

Creativity and randomness

Neurons arrangement for forming concepts

Self-improvement

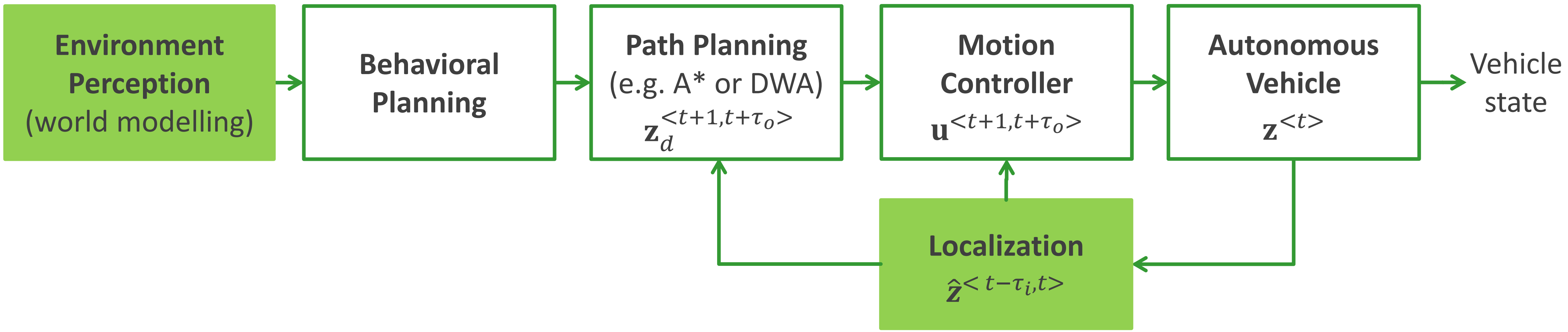
Measure problem complexity

Traditional *Perception-Plan-Act* Pipeline

Learning-based modules
Non-learning modules

Perception and Control components are treated independently of each other

- Perception and path planning are decoupled from the motion controller
- **Advantages**: reduced design complexity due to modularization
- **Disadvantages**:
  - Disturbances and intrinsic components dependencies are not taken into account
  - If one component fails (e.g. path planning), the entire control system will fail



Traditional *Perception-Plan-Act* Pipeline

Learning-based modules

Non-learning modules

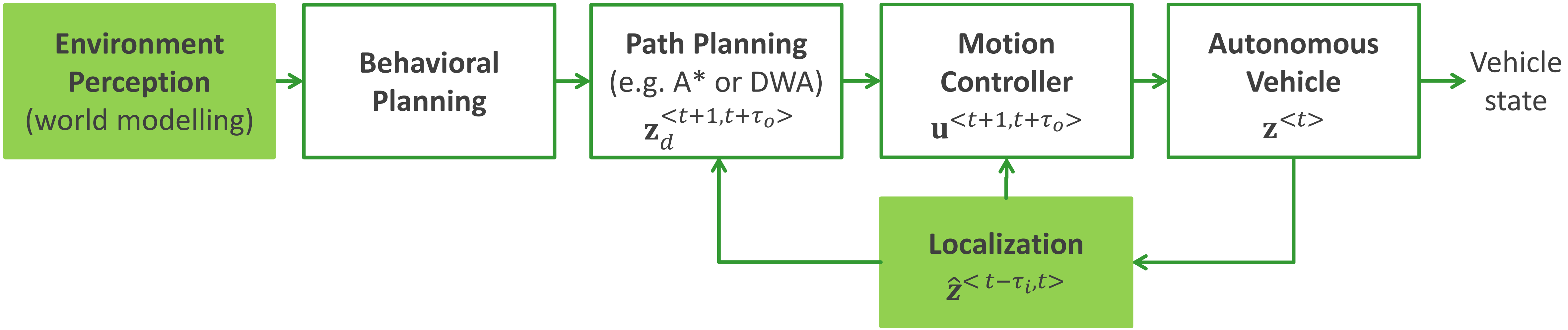
Perception and Control components are treated independently of each other

Visual Perception Design

- **Computation Intelligence** community
- **Goal**: transform the environment in a machine understandable form

Control System Design

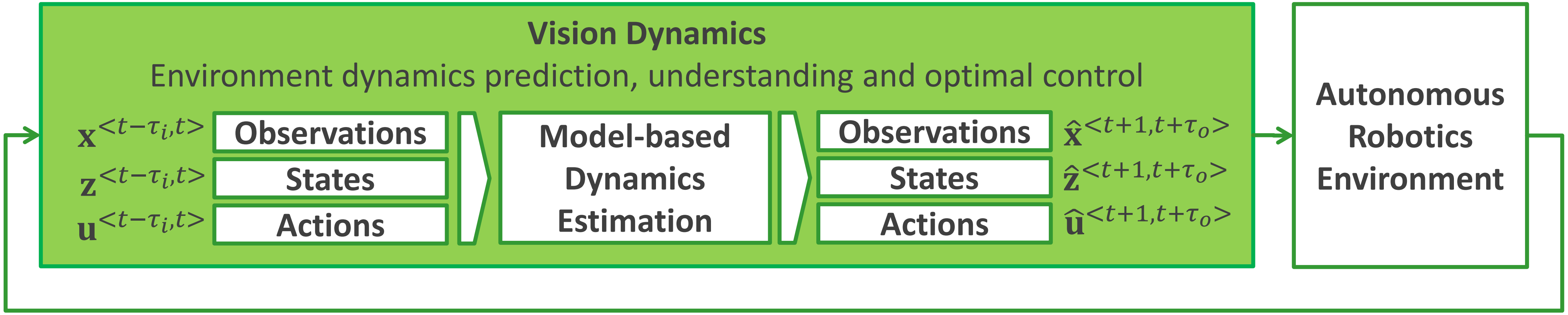
- **Automatic Control** community
- **Goal**: compute optimal control actions, based on a virtual representation of the environment



Vision Dynamics Framework

Environment perception and control based on analytical and statistical models

- Predict future **observations** $\hat{\mathbf{x}}^{<t+1,t+\tau_o>}$, **state trajectories** $\hat{\mathbf{z}}^{<t+1,t+\tau_o>}$ and **control inputs** $\hat{\mathbf{u}}^{<t+1,t+\tau_o>}$
- Optimize control inputs over prediction horizon $[t+1, t+\tau_o]$



- Assumptions:
  - The observations, states and actions are continuous and sampled at discrete time
  - The dynamics are governed by the physical laws of classical mechanics

- t – discrete sampling time
- τ_i – past sampling time horizon
- τ_o – prediction horizon

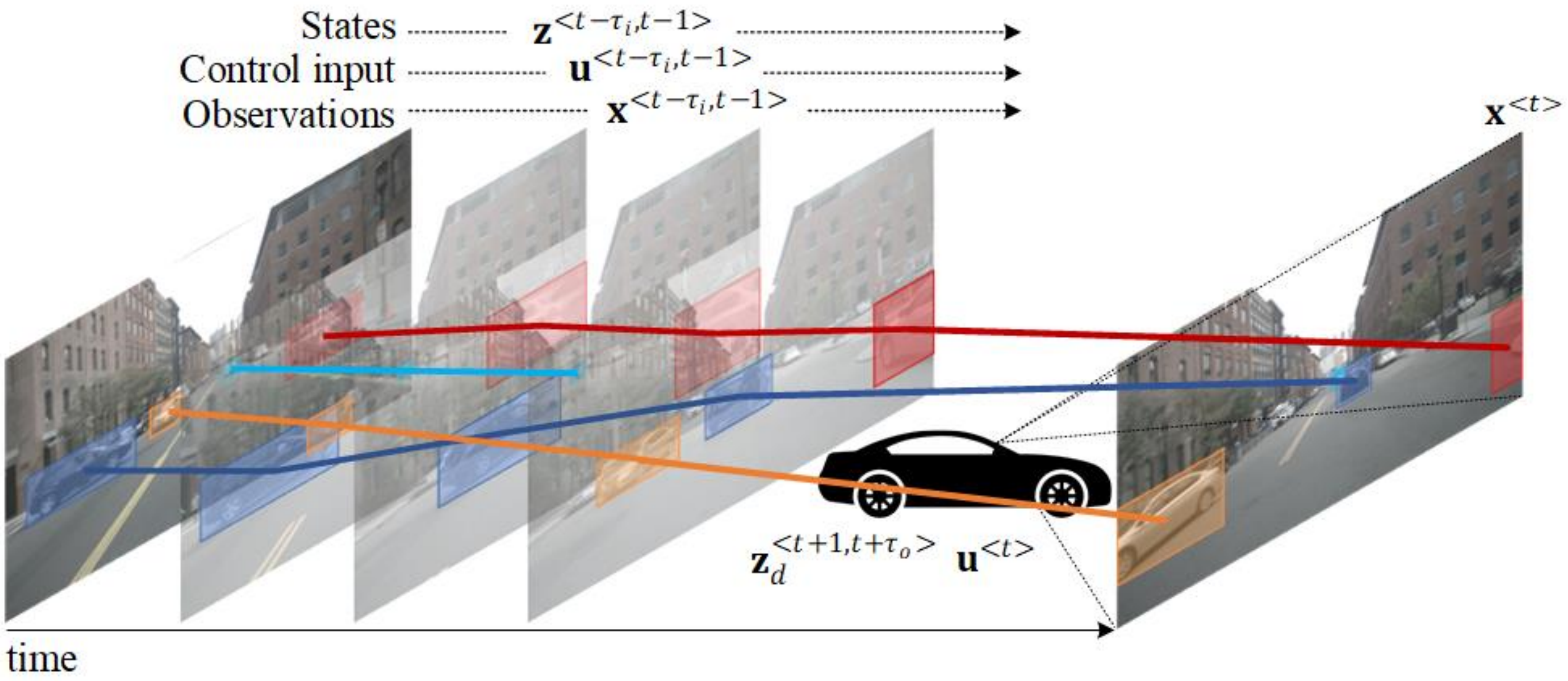
Grigorescu, “Vision Dynamics Based Learning Control”, *Learning Control*, Elsevier, 2020 (**to be published**).

Elektrobit

Vision Dynamics Approach

Environment perception and control based on analytical and statistical models

- Predict future **observations** $\hat{\mathbf{x}}^{<t+1,t+\tau_o>}$, **state trajectories** $\hat{\mathbf{z}}^{<t+1,t+\tau_o>}$ and **control inputs** $\hat{\mathbf{u}}^{<t+1,t+\tau_o>}$
- Optimize control trajectories over prediction horizon $[t+1, t+\tau_o]$



Vision Dynamics Approach

τ_i – past sampling time horizon (input)

τ_o – prediction horizon (output)

State transition modelling

given (a set of observations $\mathbf{x}$, states $\mathbf{s}$ and actions $\mathbf{a}$)

$$\mathbf{x}^{<t-\tau_i,t>}, \mathbf{s}^{<t-\tau_i,t>}, \mathbf{a}^{<t+1,t+\tau_o>}$$

find (a mapping)

$$h: X \times S \to A$$

such that

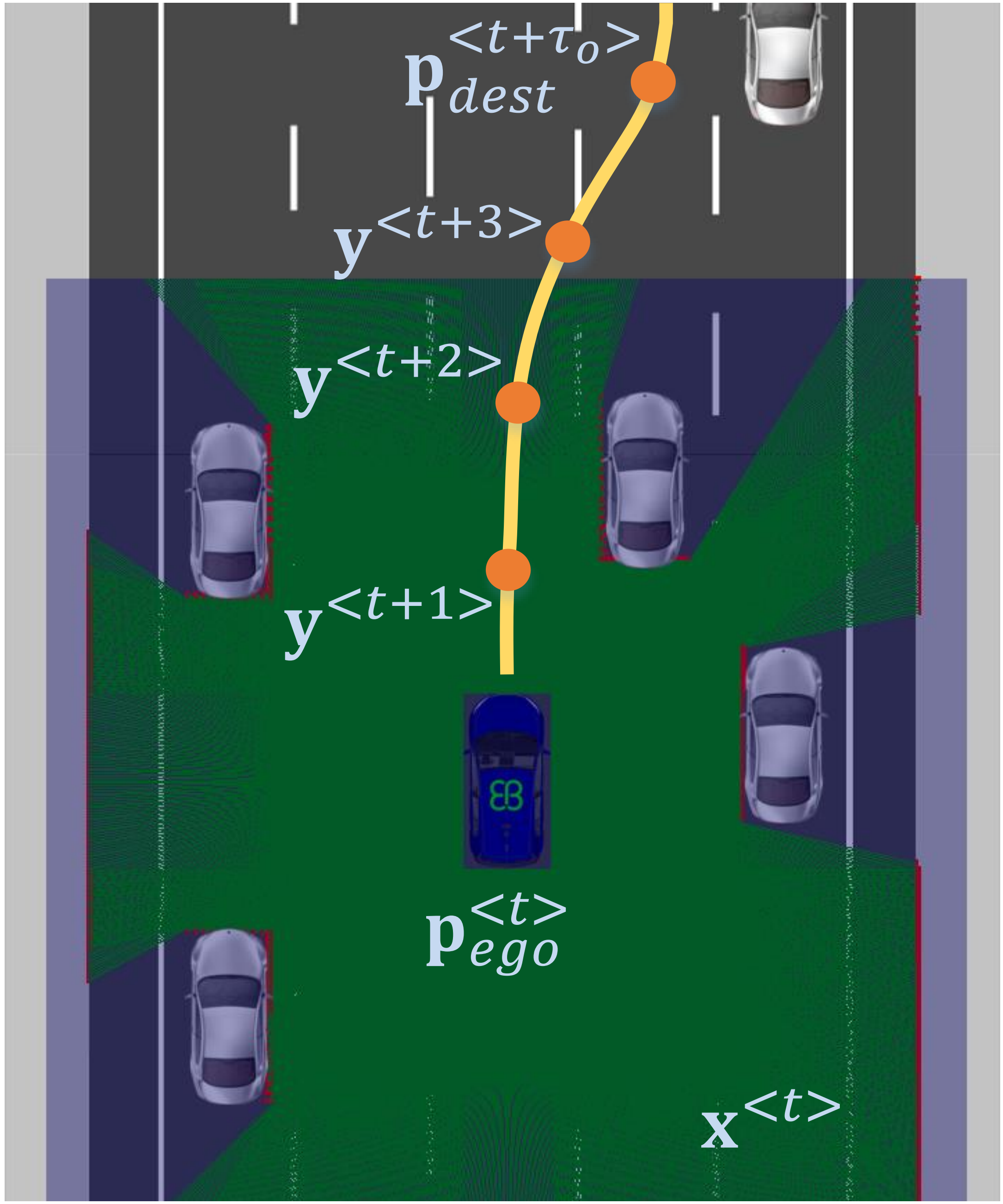
$$\mathbf{z}^{<t+1>} = \underbrace{f(\mathbf{z}^{<t>}, \mathbf{u}^{<t>})}_{\substack{a-priori \\ \text{model}}} + \boxed{\underbrace{h(\mathbf{s}^{<t-\tau_i,t>})}_{\substack{\text{learned} \\ \text{statistical} \\ \text{model}}}}$$

encodes the scene‘s temporal dynamics

Autonomous Driving Problem

A vision dynamics control perspective

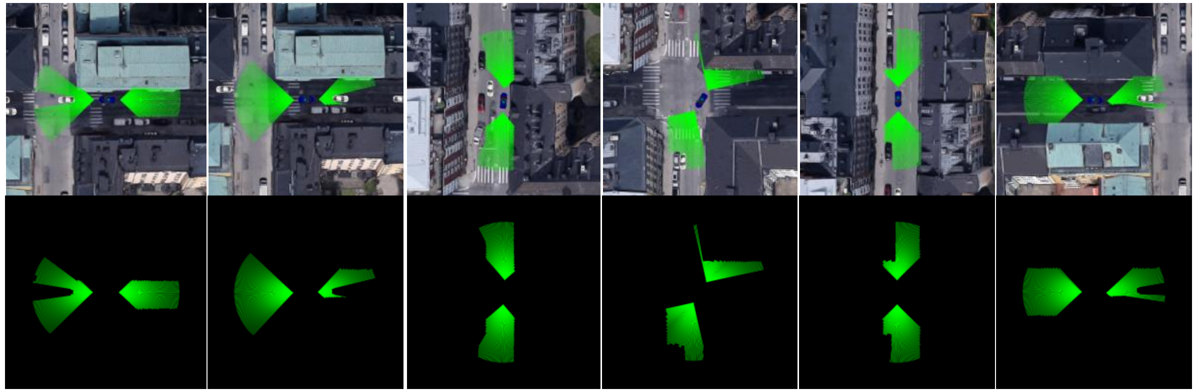
- *Given*
  - a sequence of past occupancy grid observations $\mathbf{X}^{<t-\tau_i,t>} = [\mathbf{x}^{<t-\tau_i>}, \dots, \mathbf{x}^{<t-1>}, \mathbf{x}^{<t>}]$
  - the position of the ego-vehicle $\mathbf{p}_{ego}^{<t>} \in \mathbb{R}^2$ in occupancy grid space $\mathbf{x}^{<t>}$
  - and a destination position $\mathbf{p}_{dest}^{<t+\tau_o>}$
- *the task is to*
  - estimate a local state trajectory $\mathbf{Y}^{<t+1,t+\tau_o>} = [\mathbf{y}^{<t+1>}, \dots, \mathbf{y}^{<t+\tau_o>}]$ (yellow line), encoding position and velocity
  - to destination point $\mathbf{p}_{dest}^{<t+\tau_o>}$
  - along a prediction horizon τ_o
- *Observations:*
  - sequences of past occupancy grids $\mathbf{X}^{<t-\tau_i,t>}$, marking free-space with green and obstacles with red

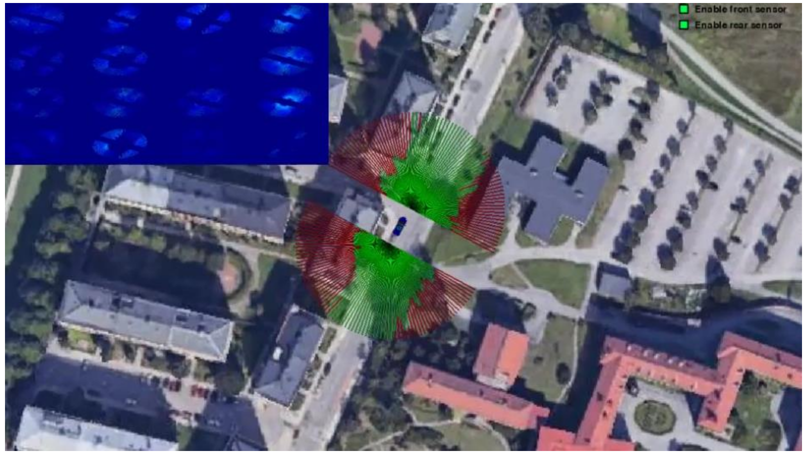


Sequences of Occupancy Grids as Observations

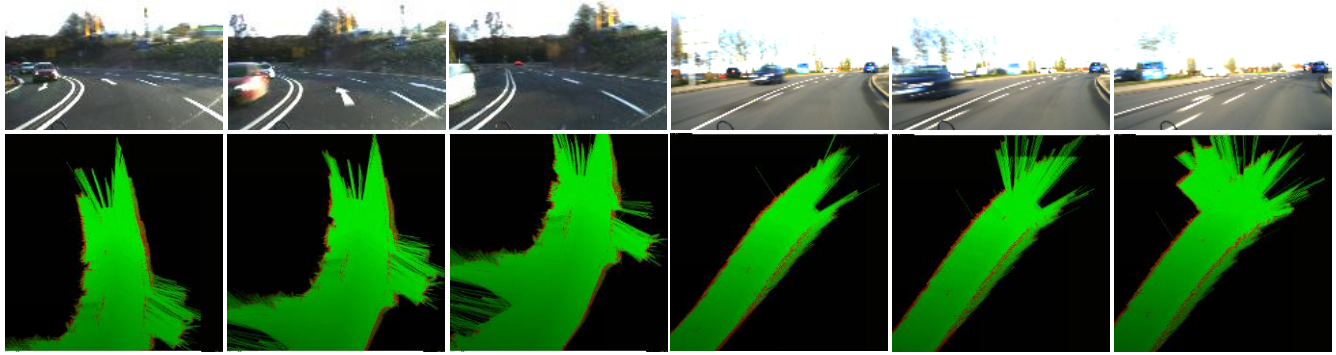
- Occupancy Grids are fused video camera, LiDAR and radar data observations (green: free space, red: obstacles, black: unknown)

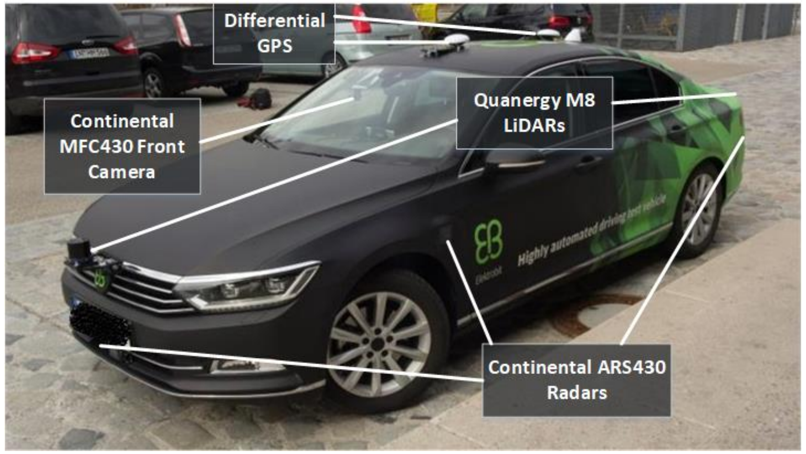
Simulated Occupancy Grids





Real-world Occupancy Grids

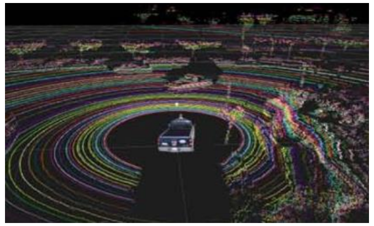




Elektrobit

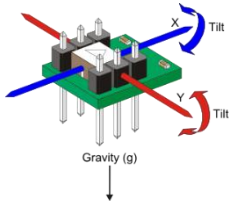
Representation Learning in Robotic Perception

Laser

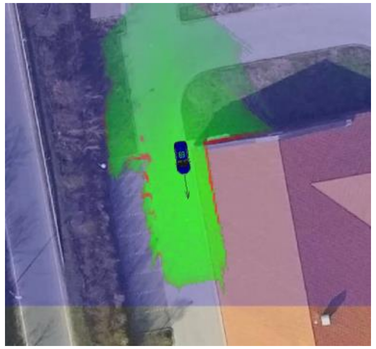


GPS

IMU

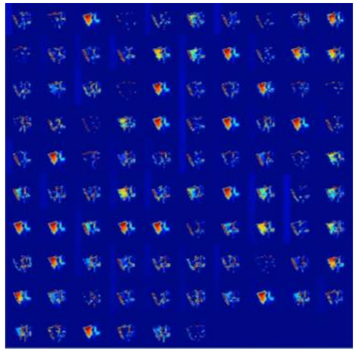


Camera



Grid Fusion Space

Representation Learning

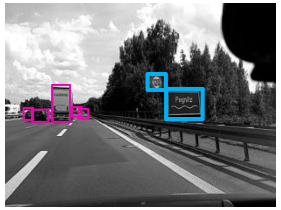


Deep Neural Networks

AI Inference Engine A

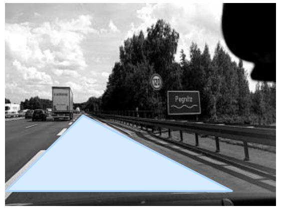
Boundaries and lanes detector

AI Inference Engine B



Driving environment recognition

AI Inference Engine C



Drivable area segmentation

⋮ ⋮

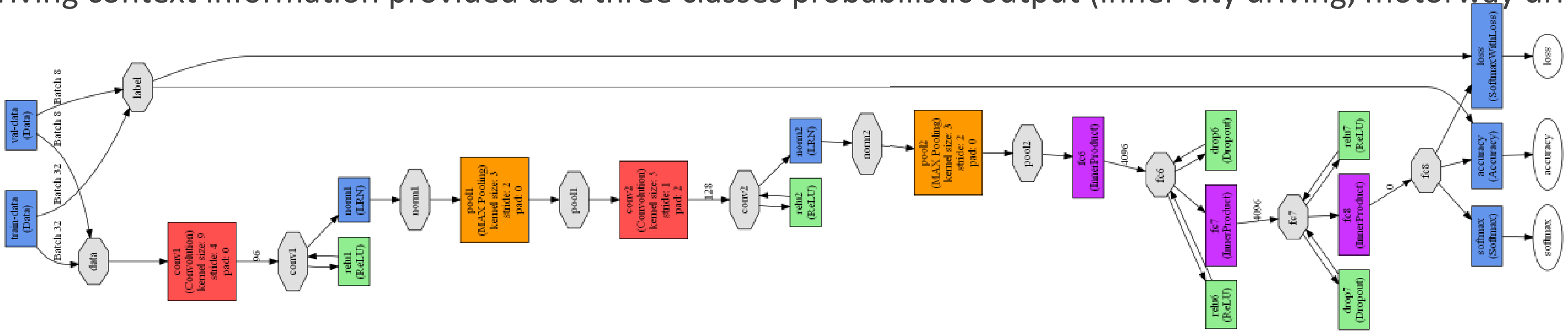
AI Inference Engine *n*

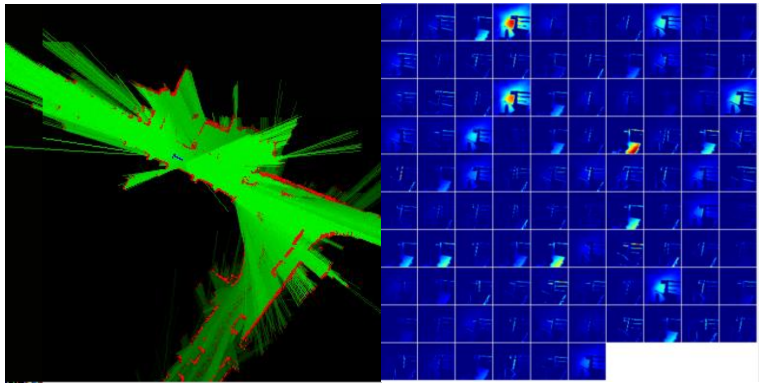
Motion planning and control

Deep Grid Net (DGN)

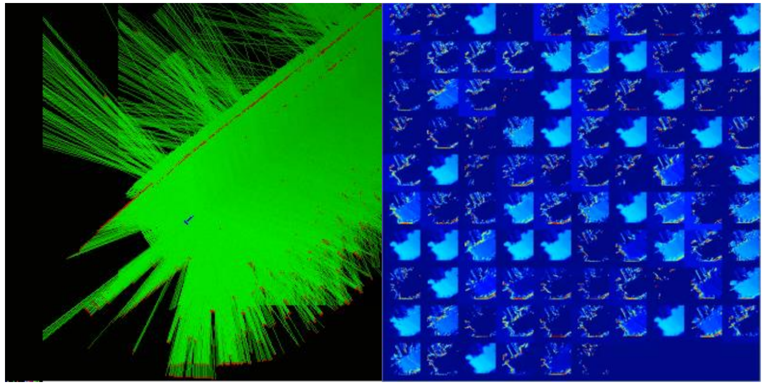
Deep Grid Net (*DGN*): A Neural Network for Driving Context Understanding

- **Input**: Grid data acquired from different driving scenarios
- **Output**: Driving context information provided as a three classes probabilistic output (inner city driving, motorway driving and parking)

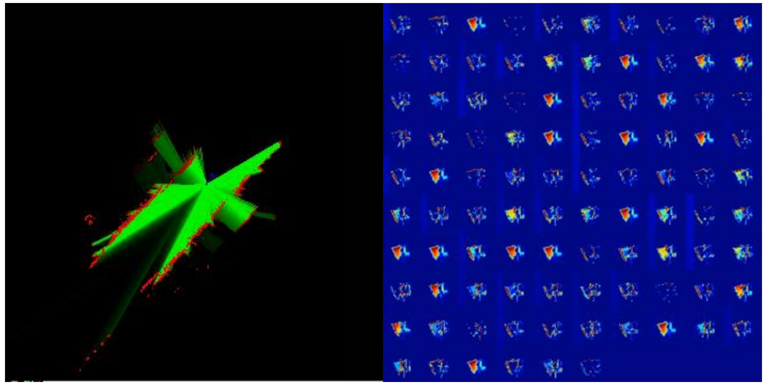




Inner city



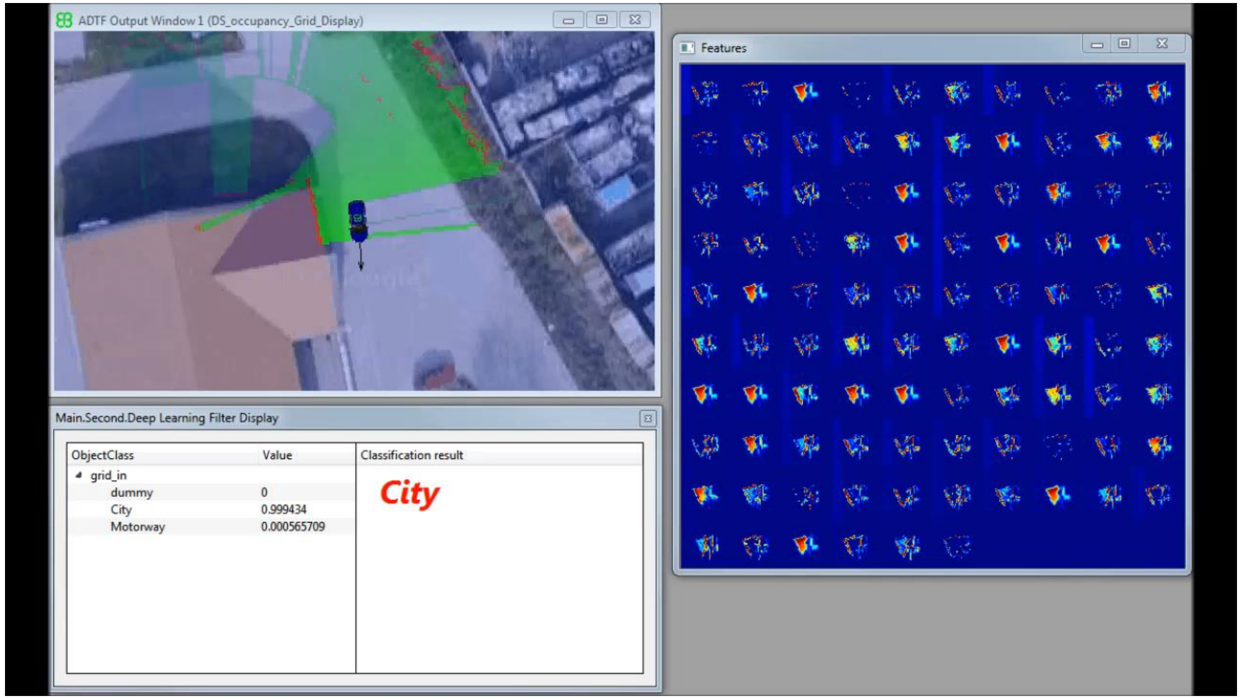
Motorway

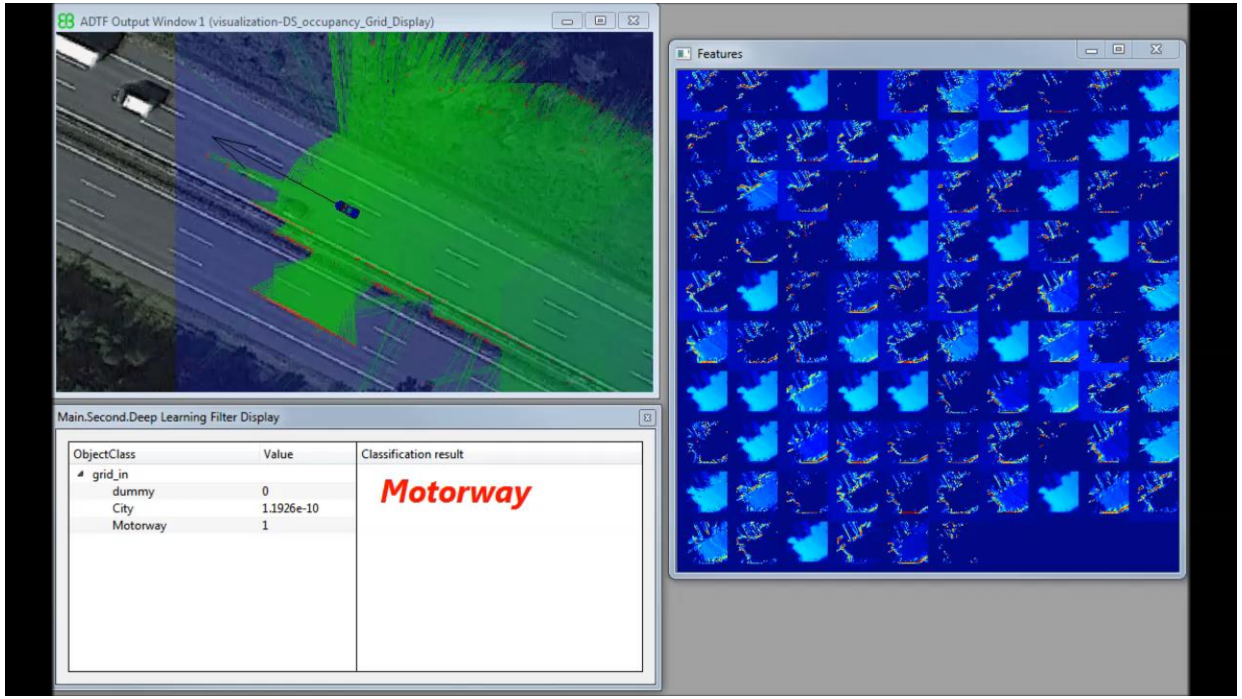


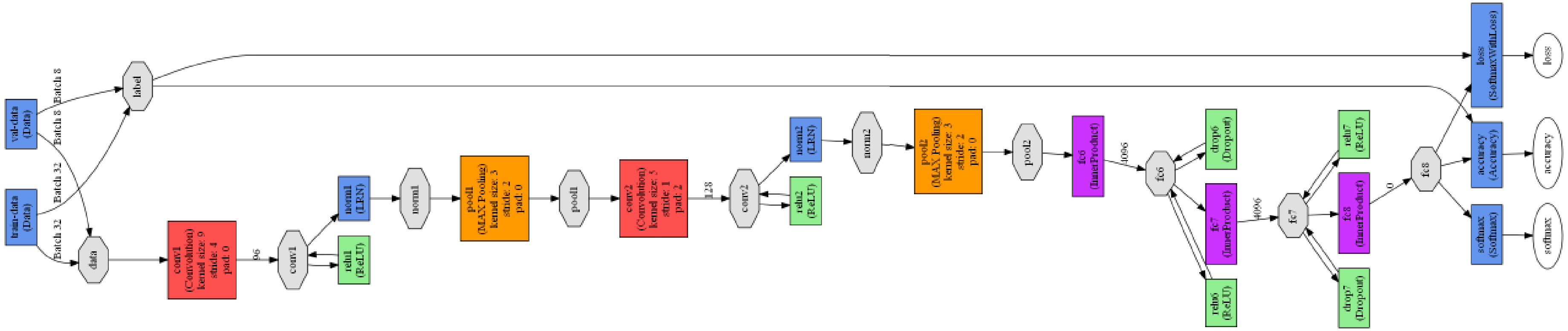
Parking place

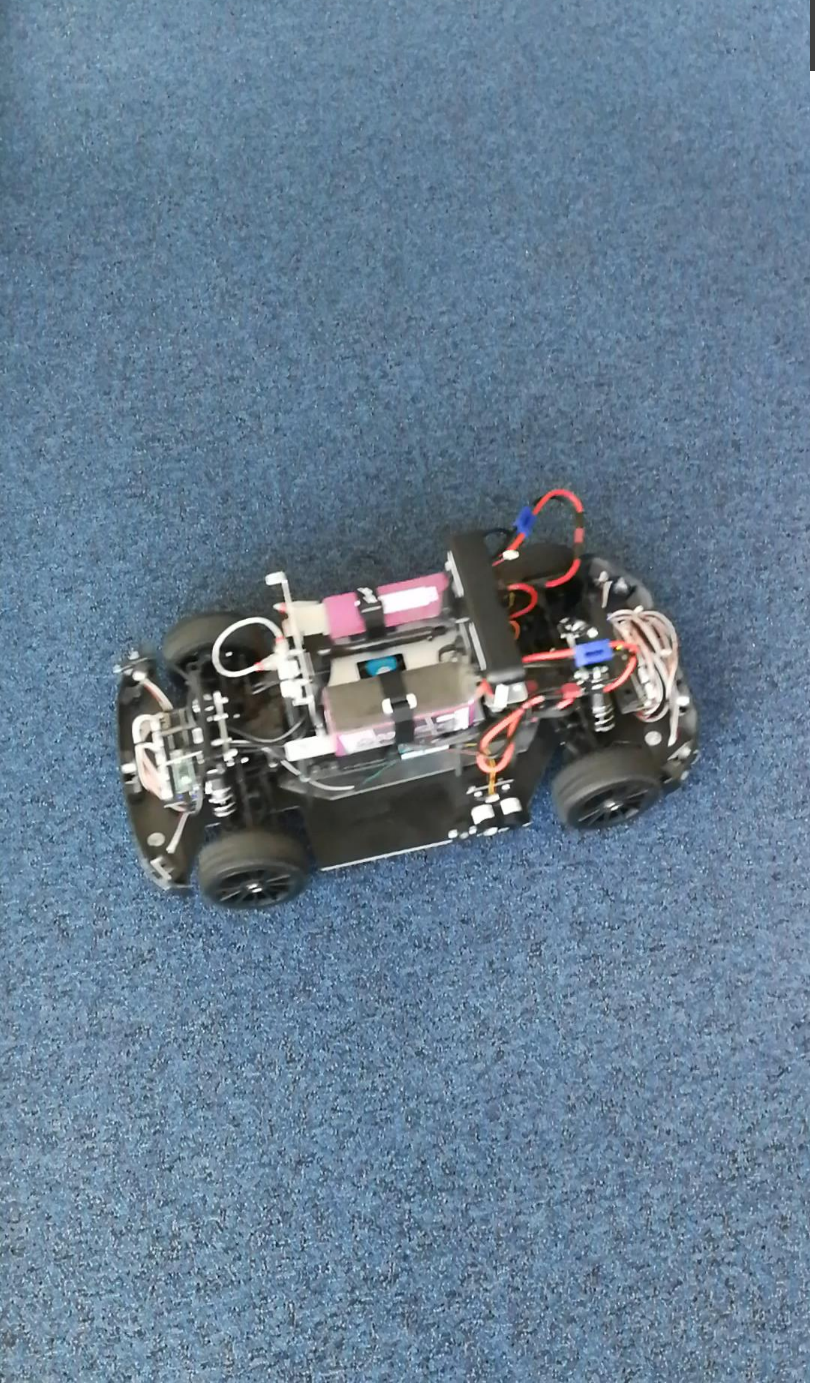
Marina et. all, "Deep Grid Net (DGN): A Deep Learning System for Real-Time Driving Context Understanding", Int. Conf. on Robotic Computing IRC 2019, Naples, Italy, February 25-27, 2019.

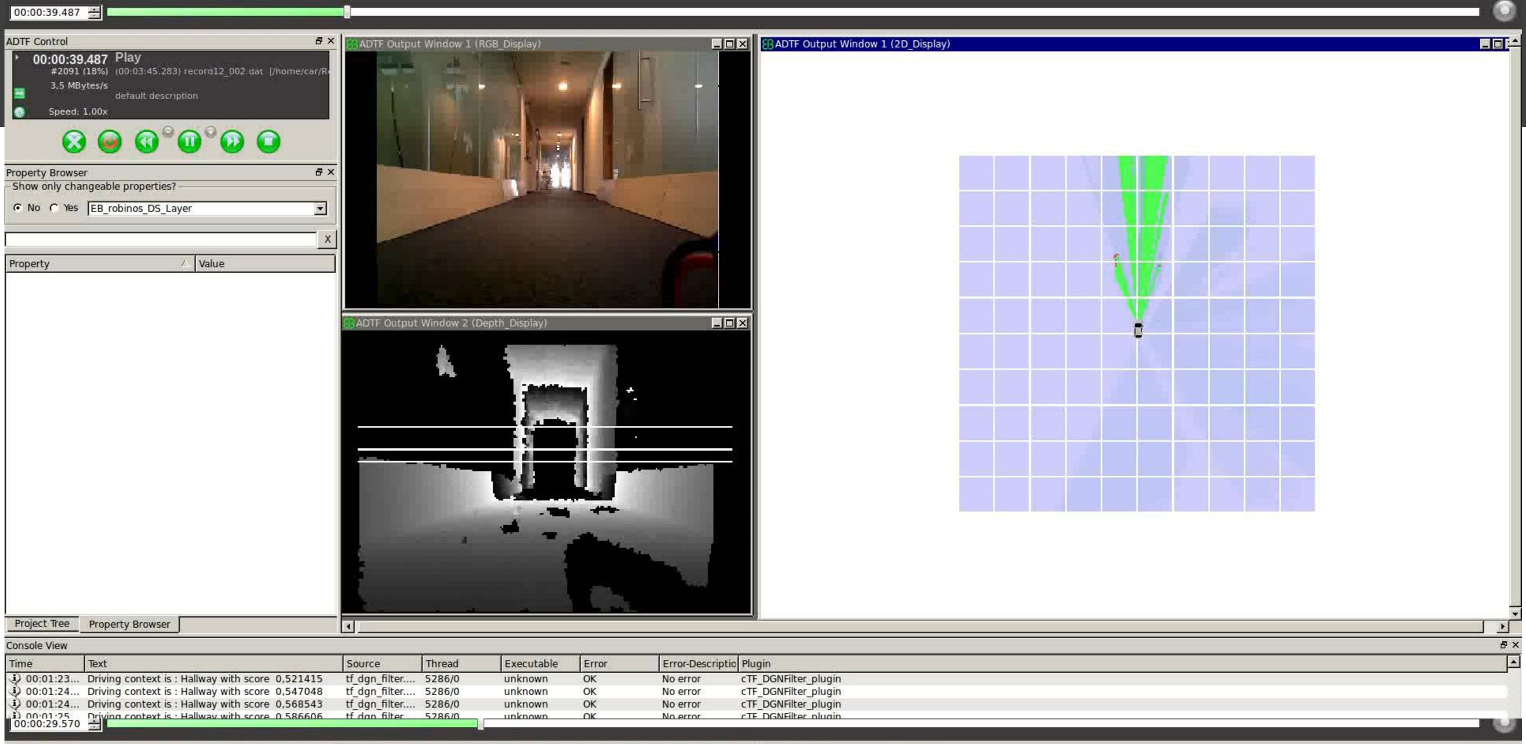
Deep Grid Net (DGN)

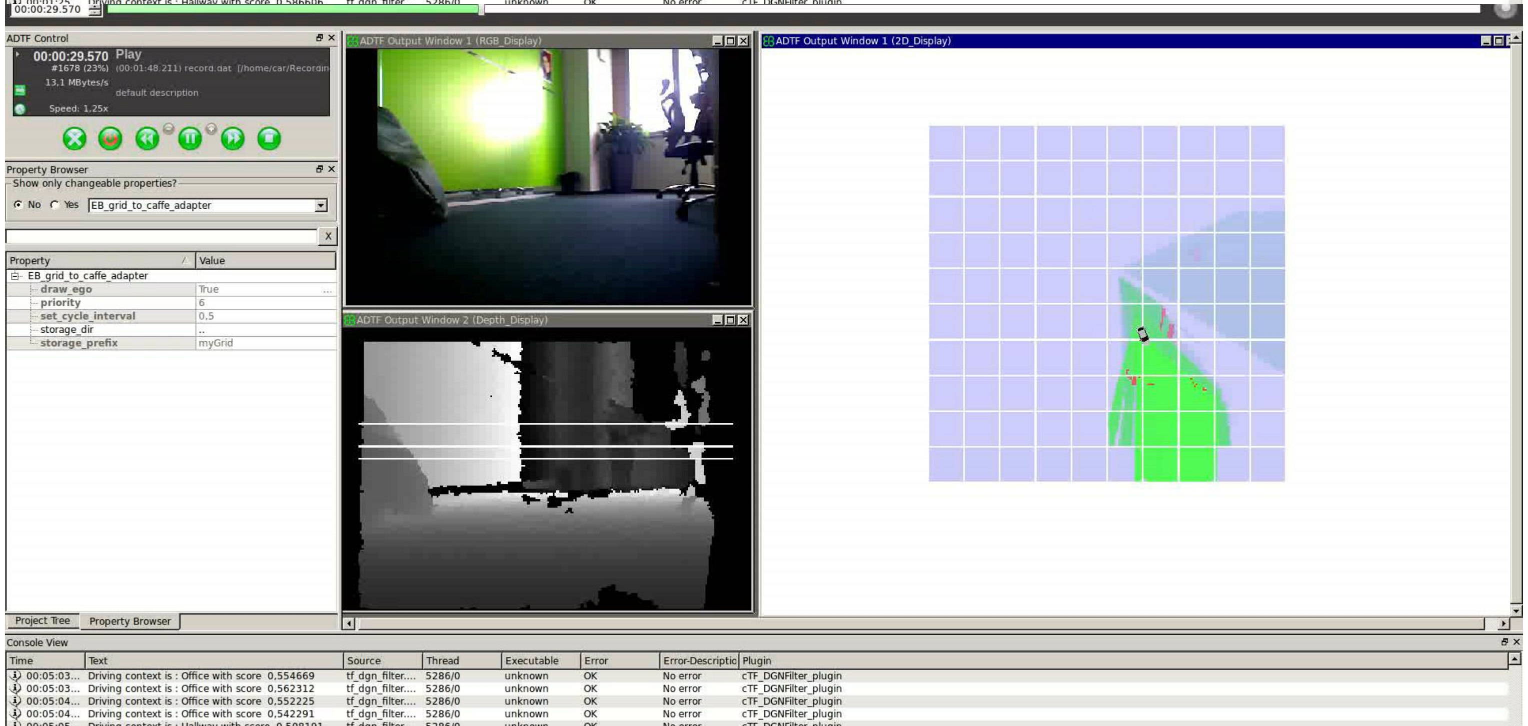






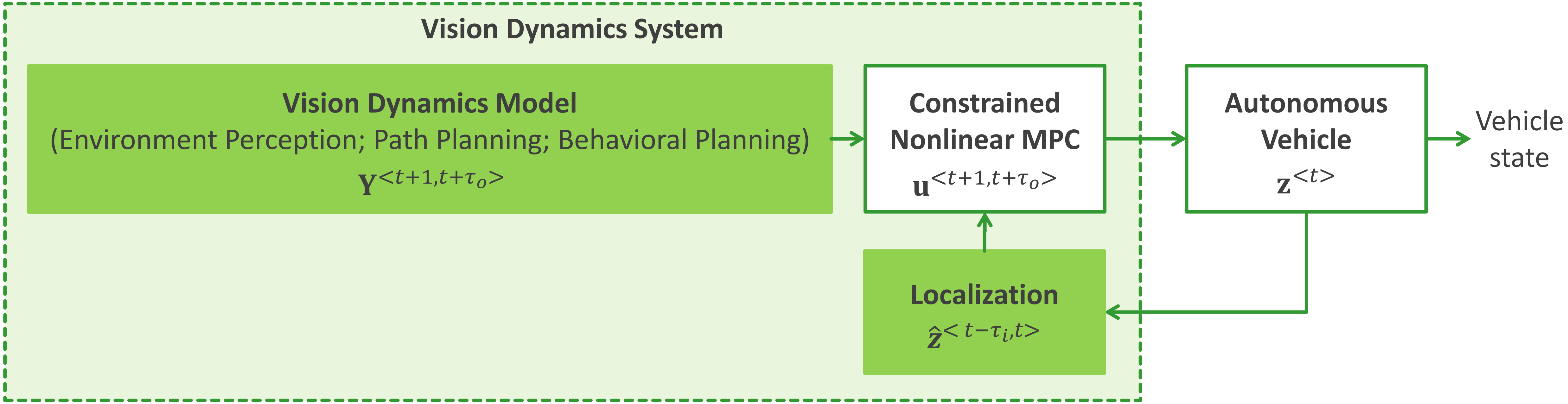



00:00:39.487
ADTF Control
00:00:39.487 Play
#2091 (18%) (00:03:45.283) record12_002.dat [/home/car/R
3,5 MBytes/s
default description
Speed: 1.00x
ADTF Output Window 1 (RGB_Display)
ADTF Output Window 2 (Depth_Display)
ADTF Output Window 1 (2D_Display)
Property Browser
Show only changeable properties?
No
Yes
EB_robinos_DS_Layer
Property
Value
Project Tree
Property Browser
Console View
Time
Text
Source
Thread
Executable
Error
Error-Descriptio
Plugin
00:01:23.... Driving context is : Hallway with score 0,521415 tf_dgn_filter.... 5286/0 unknown OK No error cTF_DGNFilter_plugin
00:01:24.... Driving context is : Hallway with score 0,547048 tf_dgn_filter.... 5286/0 unknown OK No error cTF_DGNFilter_plugin
00:01:24.... Driving context is : Hallway with score 0,568543 tf_dgn_filter.... 5286/0 unknown OK No error cTF_DGNFilter_plugin


00:00:29.570
ADTF Control
00:00:29.570 Play
#1678 (23%) (00:01:48.211) record.dat [/home/car/Recordin
13,1 MBytes/s
default description
Speed: 1.25x
ADTF Output Window 1 (RGB_Display)
ADTF Output Window 2 (Depth_Display)
ADTF Output Window 1 (2D_Display)
Property Browser
Show only changeable properties?
No
Yes
EB_grid_to_caffe_adapter
Property
Value
EB_grid_to_caffe_adapter
draw_ego True
priority 6
set_cycle_interval 0,5
storage_dir ..
storage_prefix myGrid
Project Tree
Property Browser
Console View
Time
Text
Source
Thread
Executable
Error
Error-Descriptio
Plugin
00:05:03.... Driving context is : Office with score 0,554669 tf_dgn_filter.... 5286/0 unknown OK No error cTF_DGNFilter_plugin
00:05:03.... Driving context is : Office with score 0,562312 tf_dgn_filter.... 5286/0 unknown OK No error cTF_DGNFilter_plugin
00:05:04.... Driving context is : Office with score 0,552225 tf_dgn_filter.... 5286/0 unknown OK No error cTF_DGNFilter_plugin
00:05:04.... Driving context is : Office with score 0,542291 tf_dgn_filter.... 5286/0 unknown OK No error cTF_DGNFilter_plugin

NeuroTrajectory: Perception-Planning Deep Network

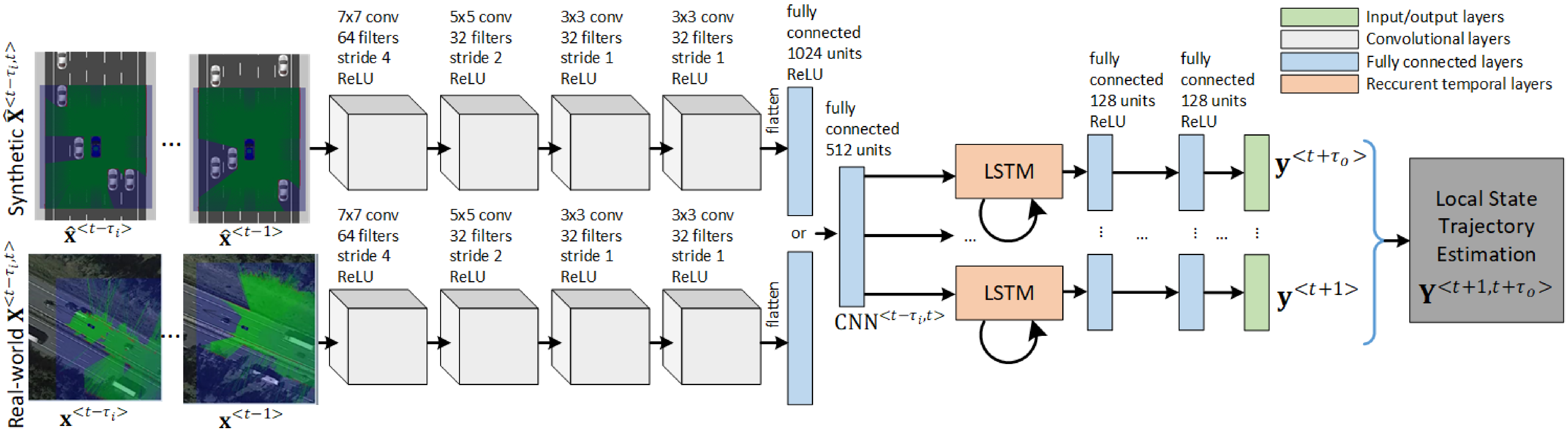
- Local state prediction
- Encode environment perception, path planning and behavioral planning within a single statistical model (DNN)



Grigorescu et. all, "NeuroTrajectory: A Neuroevolutionary Approach to Local State Trajectory Learning for Autonomous Vehicles", *IEEE Robotics and Automation Letters*, 2019.

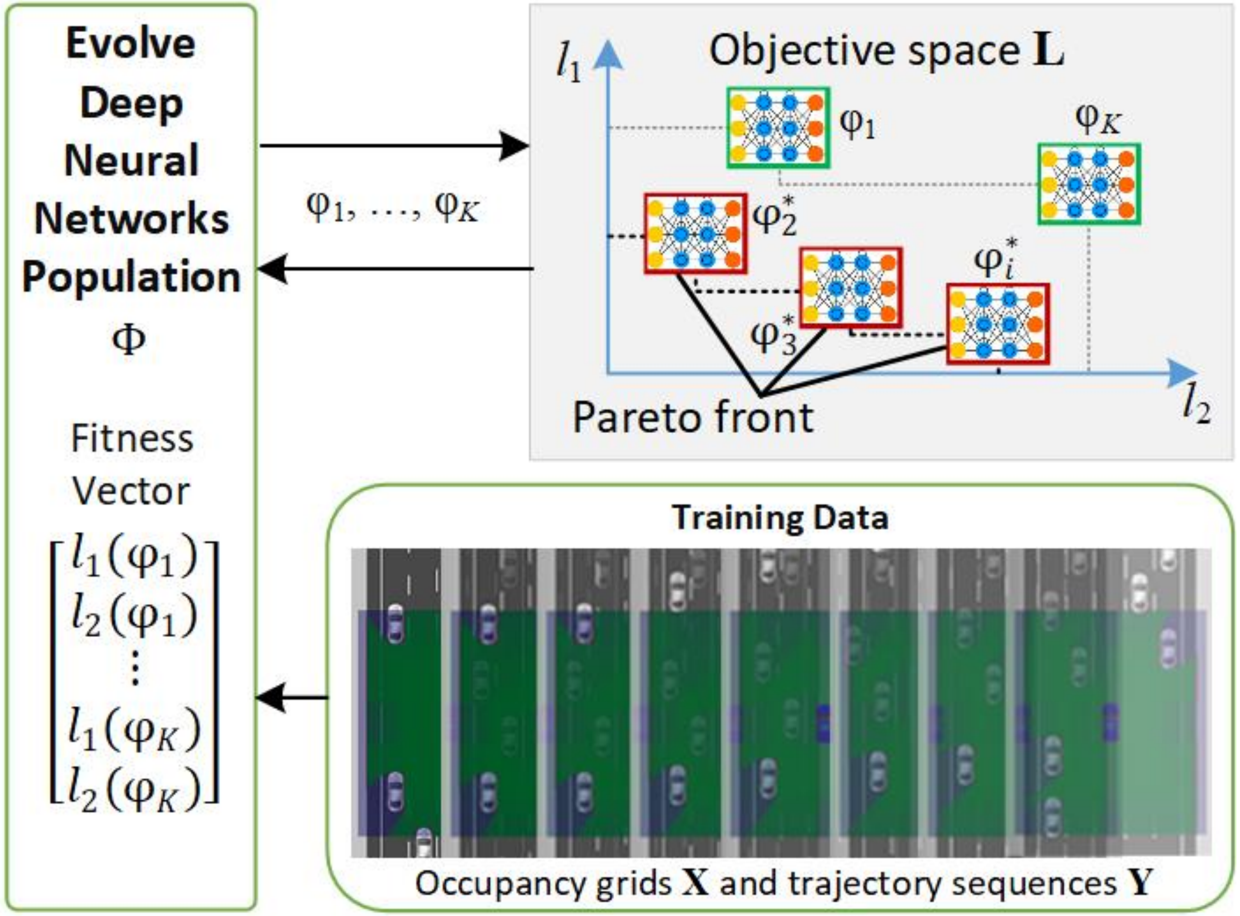
NeuroTrajectory: Vision Dynamics Model

- $\text{CNN}^{<t-\tau_i,t>}$ sequence of spatial features is further fed to a stack of LSTM branches
- $\mathbf{Y}^{<t+1,t+\tau_o>} = [\mathbf{y}^{<t+1>}, \dots, \mathbf{y}^{<t+\tau_o>}]$ - estimated local states along prediction interval $[t+1, t+\tau_o]$
- $\mathbf{y}^{<t+i>}$ - local state element predicted by LSTM branch i



NeuroTrajectory: Evolutionary DNN Training

- Multi-objective training on *Objective Space L*: evolving a population of deep neural networks φ_K
- Multi-objective loss: the ego-vehicle's *i*) **traveled path $\boldsymbol{l_1}$**, *ii*) **lateral velocity $\boldsymbol{l_2}$** and *iii*) **longitudinal velocity $\boldsymbol{l_3}$**



- $M = (S, A, T, L)$
- $S -$ states trajectories $\mathbf{s}^{<t-\tau_i,t>} = (\mathbf{x}^{<t-\tau_i,t>})$
- A – trajectory sequences; $\mathbf{Y}^{<t+1,t+\tau_o>} \in A$
- $T: S \times A \times S \to [0,1]$ – transition function describing the probability of arriving in state $\mathbf{s}^{<t+\tau_o>}$, after optimizing over trajectory $\mathbf{a}^{<t>}$
- $L: S \times A \times S \to \mathbb{R}^n$ – multi-objective cost function quantifying the quality of trajectory $\mathbf{Y}^{<t>}$

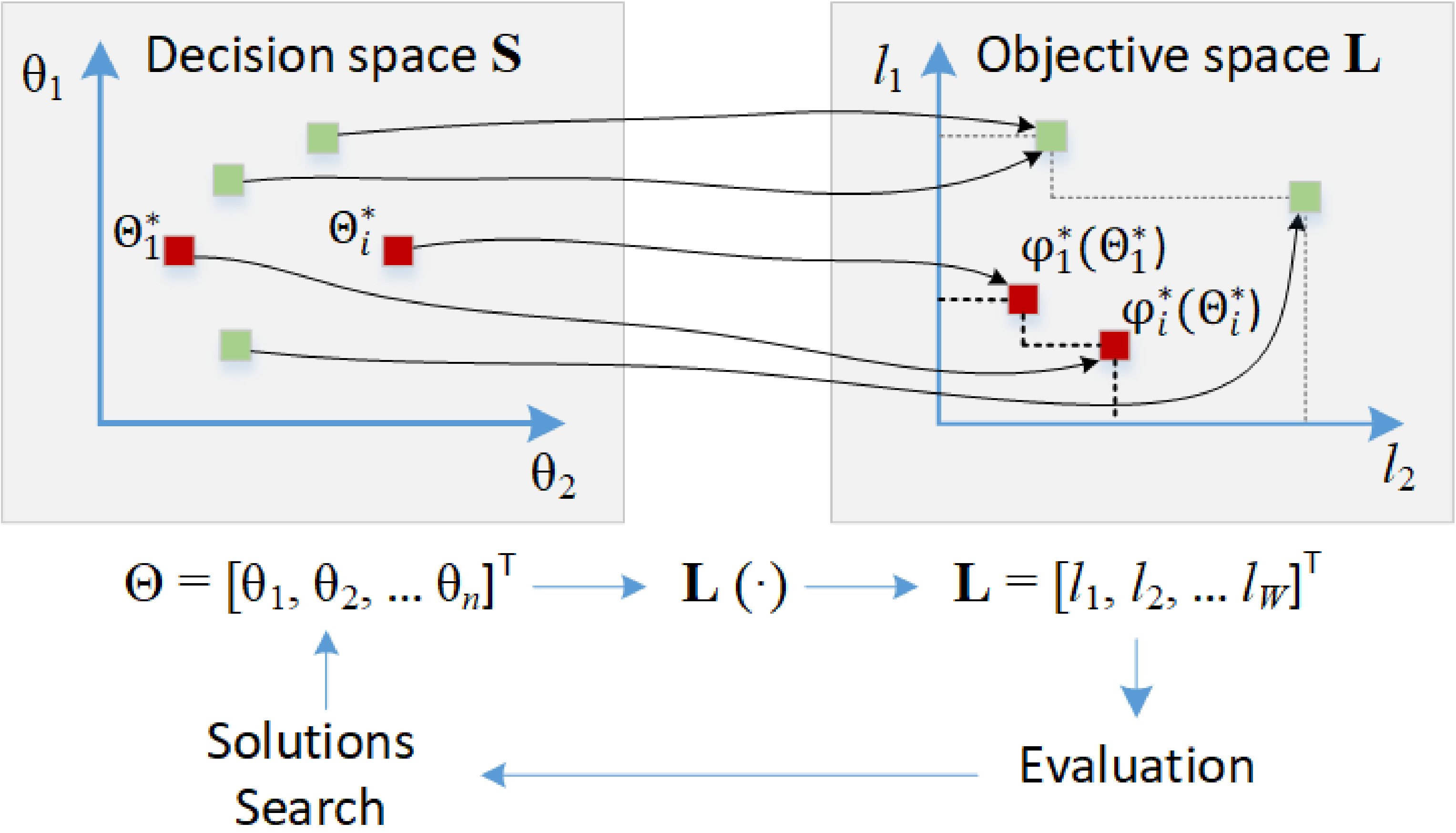
$$l_1^{<t+\tau_o>} = \sum_{i=1}^{\tau_o} \left\| \mathbf{p}_{ego}^{<t+i>} - \mathbf{p}_{dest}^{<t+i>} \right\|_2^2$$

$$l_2^{<t+\tau_o>} = \sum_{i=1}^{\tau_o} v_\delta^{<t+i>}$$

$$l_3^{<t+\tau_o>} = \sum_{i=1}^{\tau_o} v_f^{<t+i>} \in [v_{min}, v_{max}]$$

NeuroTrajectory: Evolutionary DNN Training

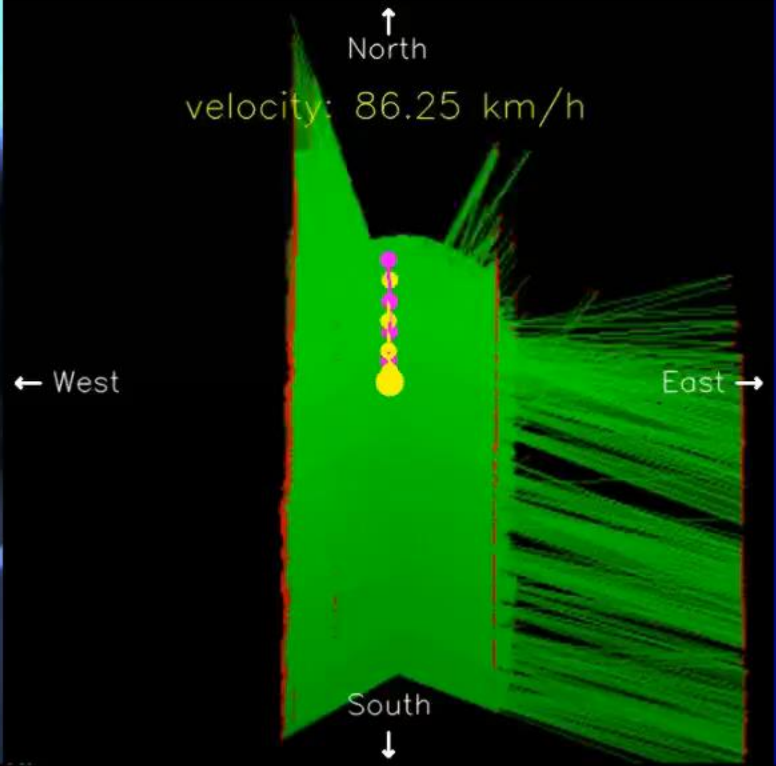
- Multi-objective training on *Objective Space L*: evolving a population of deep neural networks φ_K
- $\varphi_i(\Theta_i)$ – deep network individual with weights Θ_i
- $\Phi^* = [\varphi_1^*(\Theta_i^*), \dots, \varphi_k^*(\Theta_k^*)]$- Pareto front of optimal deep neural networks

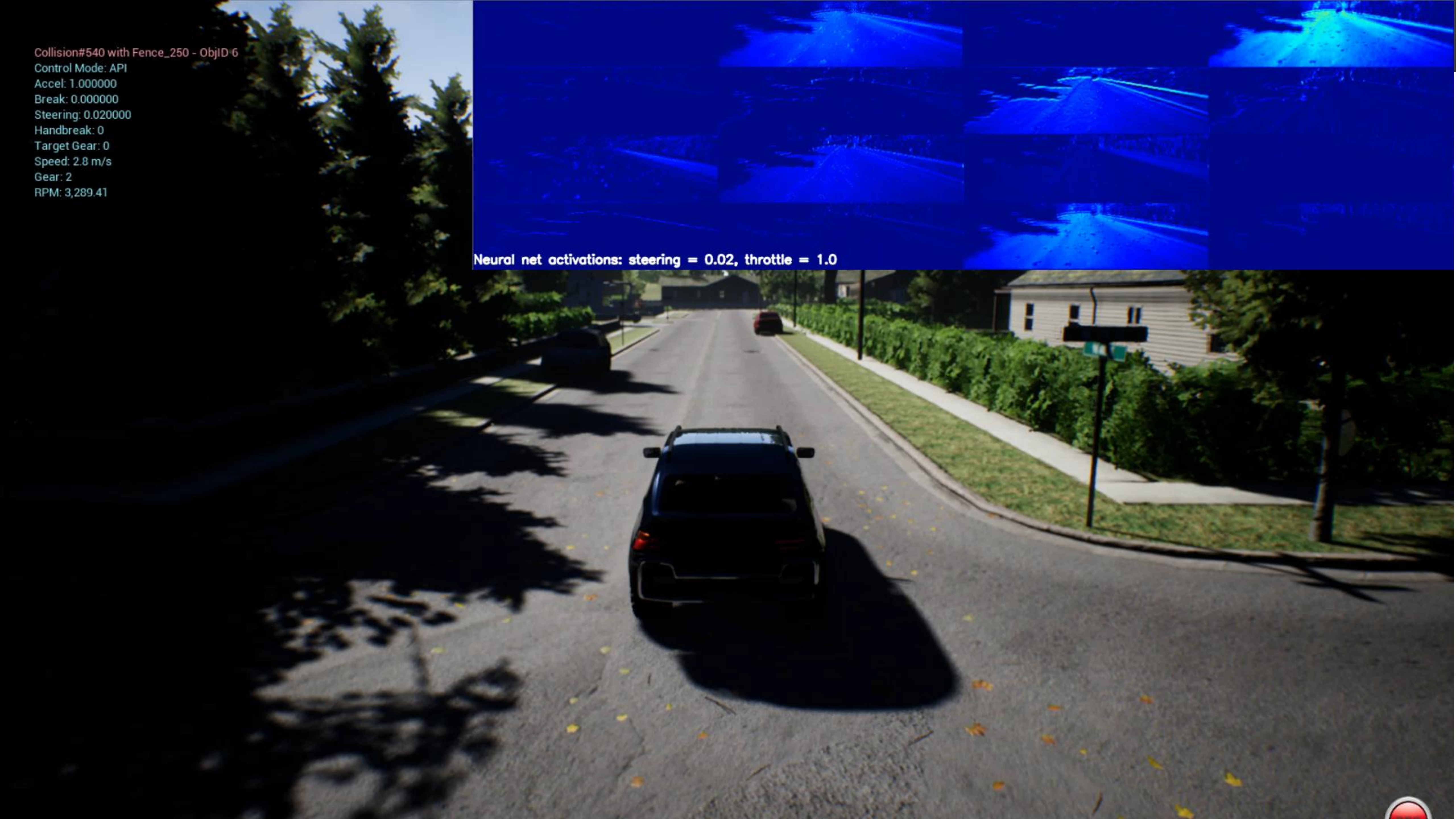



Car pos x: 89.11
Car pos y: 432.41
rel x: 89.11
rel y: 432.41
velocity: 0.0 km/h
Predicted delta values:
dx1: 0.0, dy1: 1.7
dx2: 0.0, dy2: 2.5
dx3: -0.1, dy3: 3.4
dx4: -0.1, dy4: 4.3
dx5: -0.2, dy5: 5.1
dx6: -0.2, dy6: 6.0
dx7: -0.3, dy7: 6.9
dx8: -0.4, dy8: 7.8
dx9: -0.4, dy9: 8.7
dx10: -0.5, dy10: 9.5
Enable front sensor
Enable rear sensor

Highway Driving

Reference trajectory
NeuroTrajectory




Collision#540 with Fence_250 - ObjID 6
Control Mode: API
Accel: 1.000000
Break: 0.000000
Steering: 0.020000
Handbreak: 0
Target Gear: 0
Speed: 2.8 m/s
Gear: 2
RPM: 3,289.41
Neural net activations: steering = 0.02, throttle = 1.0

Elektrobit

Thank you!

World Usability Day
November 12, 2020

Sorin.Grigorescu@elektrobit.com