Visualization and Simulation for ADAS & Autonomous Driving

Yu Huang

Yu.huang07@gmail.com

Sunnyvale, CA

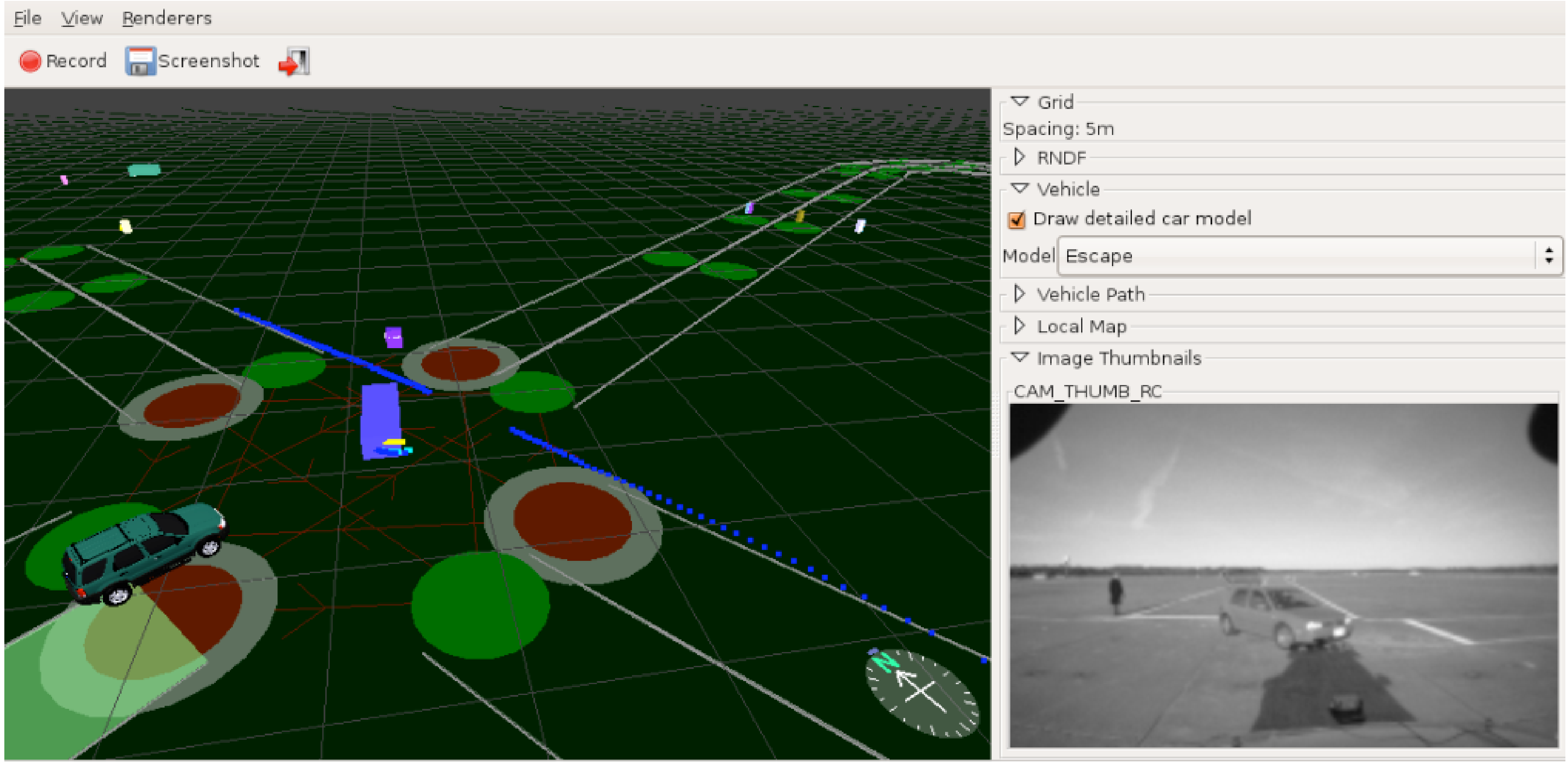
Outline

- Visualization/Simulation at DUC
- Visualization at Uber
- Simulation at Waymo
- Simulation at Daimler
- SUMO: free simulation tool
- AirSim from MS
- CARLA: An open source
- AutonoVi-Sim
- Public simulation data
  - CityScapes, Virtual Kitti, Synthia …
- OpenDRIVE
- OpenCRG
- Tass in Siemens
- IPG
- VIRES
- rFpro
- Udacity
- Righthook
- Cogna
- Cvdedia-SynCity

Boss from CMU

TROCS (Tartan Racing Operator Control Station) is an extensible GUI for developers to both monitor telemetry from Boss while it is driving and replay data offline for algorithm analysis.

Talos from MIT



Screenshot of the real-time visualization tool running “live” for an intersection testing scenario, showing RNDF and vehicle navigation info. (white, green, red), lidar (blue, yellow) and camera data, and vehicle tracker output (blue solid in intersection)

Talos from MIT

Screenshot of SimCreator simulation environ., including MIT vehicle, traffic vehicles.

Odin from Virginia Tech.

Screenshot of simulated Odin encountering a roadblock

Visualization Platform at Uber

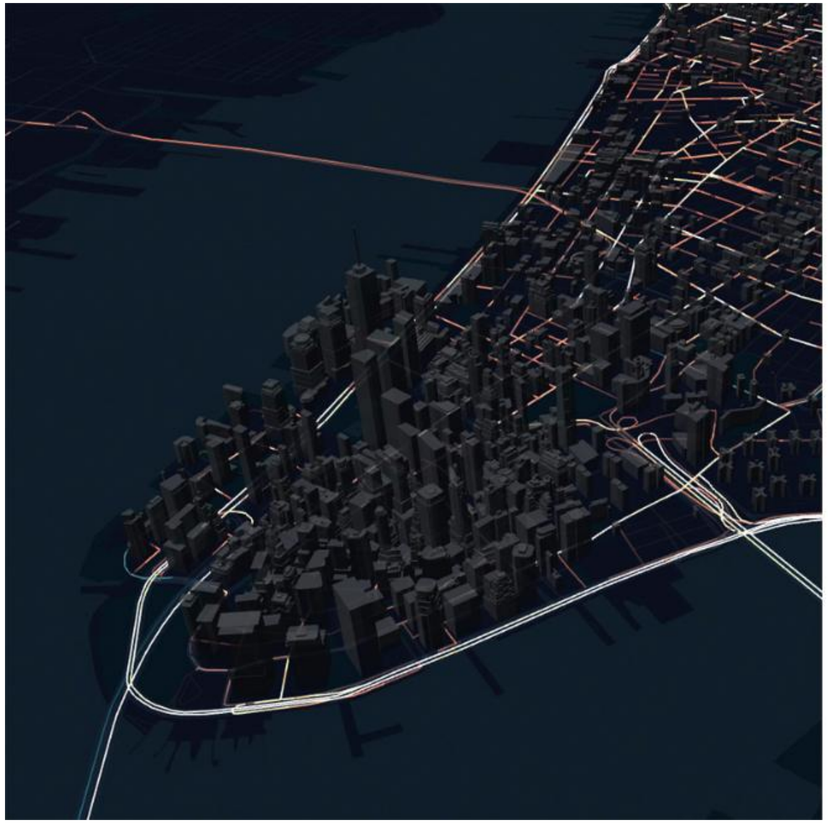
- To use visualization for exploration, inspection, debugging and exposition of data to improve how its self-driving vehicles (cars and trucks) interpret and perceive the world around;
- To understand the decisions made by an autonomous vehicle, a large amount of data is required to recreate the context around a trip.
  - This includes maps that are preprocessed and vehicle logs that are generated at runtime;
  - As HR scans of the ground surface, lane boundaries and types, turn and speed limits, and crosswalks.

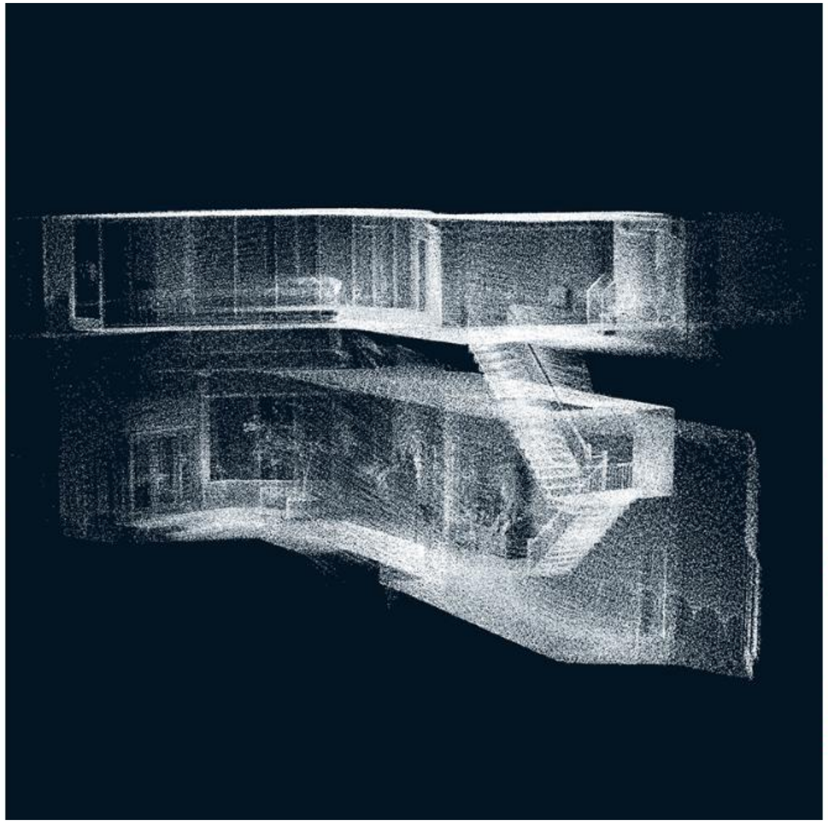
Visualization Platform at Uber

- Vehicle logs describe what the vehicle is doing, what it sees, and how it acts.
- Three critical stages are run based on sensor data: perception (measuring), prediction (forecasting), and motion planning (acting).
- A vehicle needs to be able to perceive the activity around it through its sensors, then it can predict where these objects will be in the near future, which will properly plan its next move.

Visualization Platform at Uber

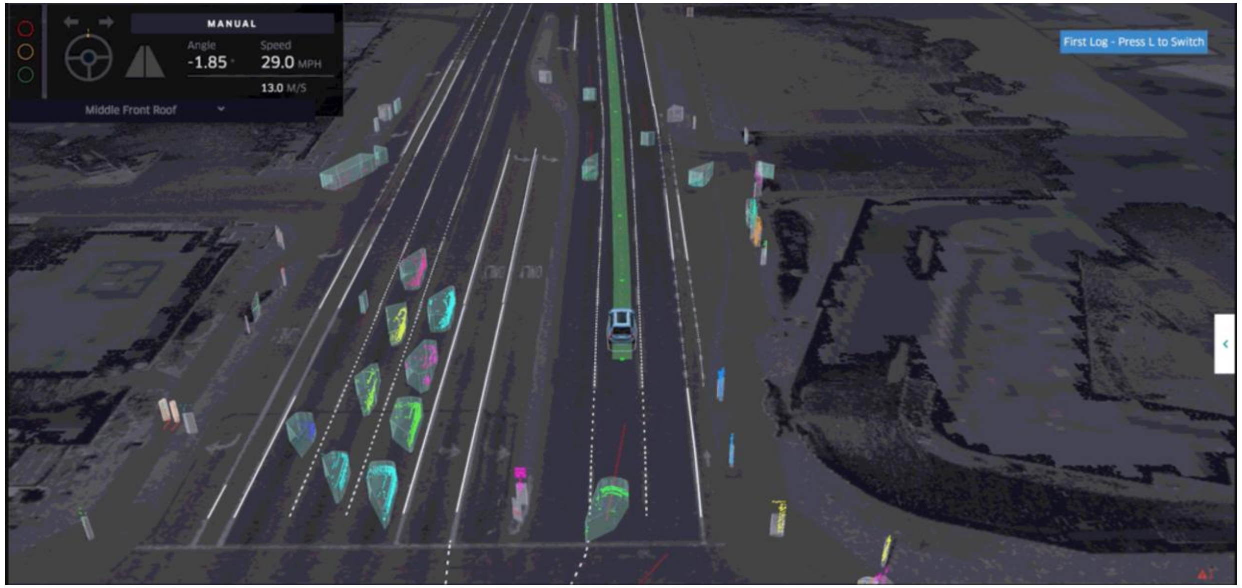
- A suite of frameworks for web-based large scale data visualization, like react-map-gl and deck.gl.
- They leverage GPU capacities in the browser to display millions of geometries at a high frame rate.
- Visualization as mapping from “bit” (data structure) to “pixel” (graphics), GPU is the ideal venue.





Visualization Platform at Uber

- deck.gl context: each layer renders a data source into a given look, either mesh (ground surfaces and cars), paths (lanes and trajectories), extruded polygons (other objects on the road), or point clouds (3D objects without current semantic meaning).
- Each layer can also specify their own coordinate system, while sharing the same camera view.
- A typical log snippet renders 60-100 layers at 30-50 frames per second.



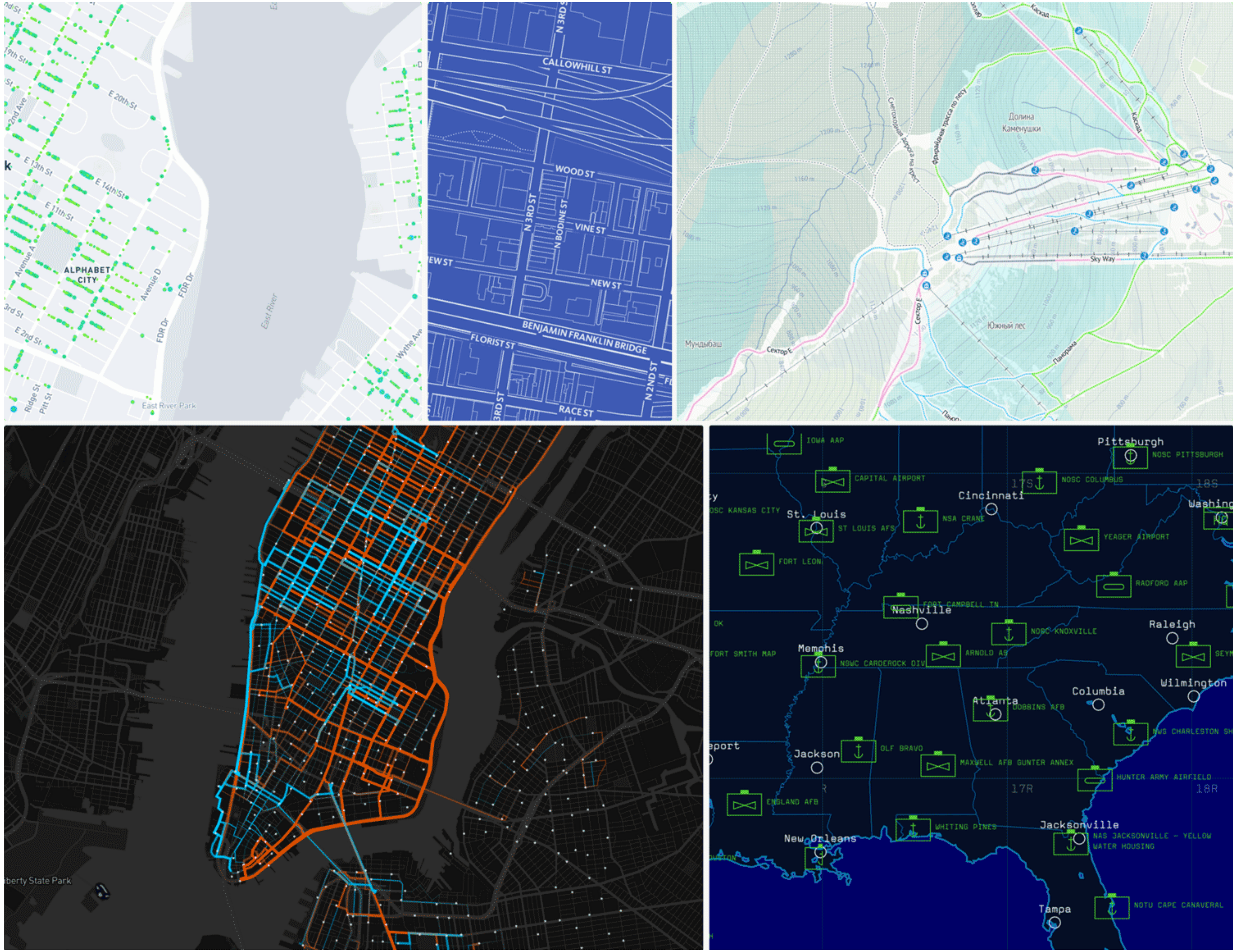
Mapbox

- Mapbox is built on **vector maps**, an advanced approach to mapping where data is delivered to the device and precisely rendered in real-time, the smooth, fast maps.
- Maps render at a super-high framerate — allowing the maps to fluidly respond to user feedback or scripted events and opening up a whole new class of apps.
  - Vector maps are roughly 1-4th the size of traditional raster implementations, which means greater performance in low-bandwidth environments and greater cost savings.
- Vector maps animate scale changes on the fly, providing smooth zooming on mobile and desktop devices.
  - To rotate the map, tilt it, and zoom in and out fluidly.
  - Icons and labels adjust to maximize legibility from any angle.
- The data for every feature in a vector map resides on the client, not the server.
  - That means data can be instantly queried, allowing for flexible map changes and user interfaces that adapt to the map automatically.
- You can customize every aspect of your map, from tweaking the colors, to hiding or showing specific layers, to choosing which information to present on your map, all while your users are interacting with the map.

Mapbox

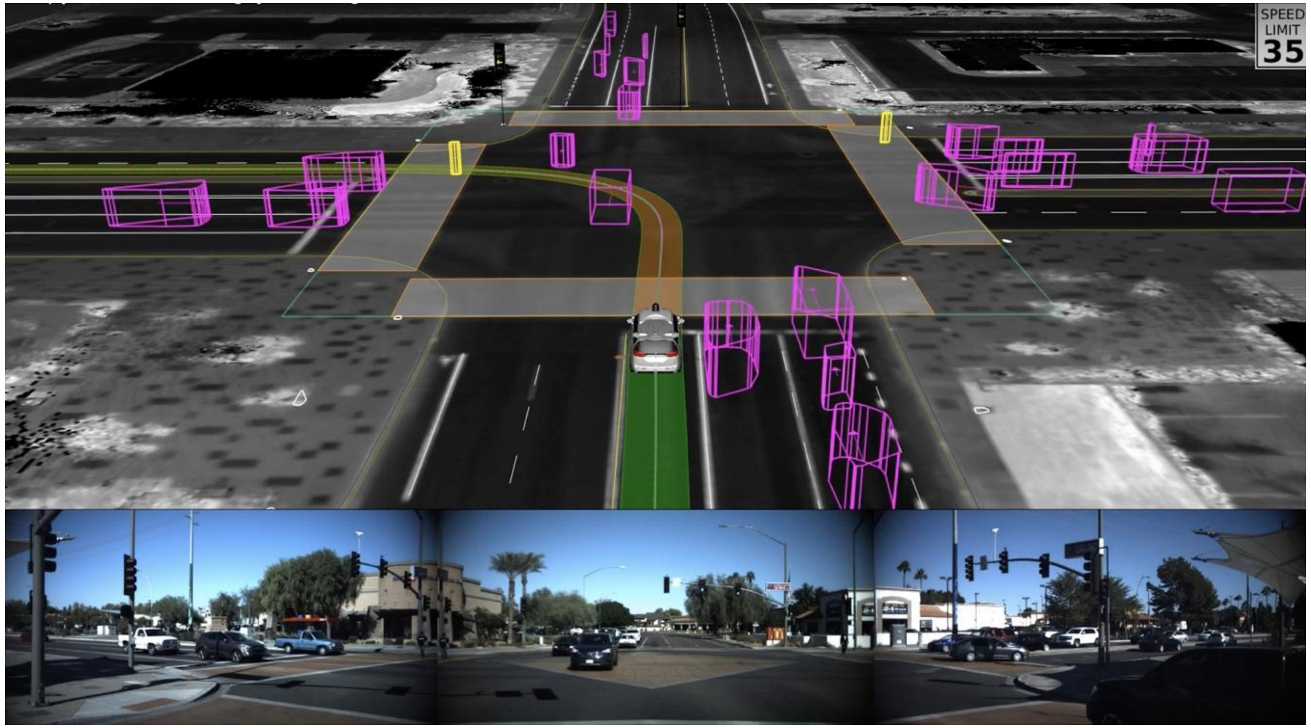
- A customizable map built for navigation use cases that updates to real world driving conditions.
- Designed for embedded auto, mobile, web with high contrast colors for day night time driving.
- Available in 2 styles, each with Day - Night versions, and fully customizable with Mapbox Studio.
- Mapbox Studio opens up complete design control, allowing to design custom vector maps.
- Take vector maps using the Mapbox iOS, Android SDKs, or add to the website using Mapbox GL JS.
- Development of vector maps and Mapbox GL: https://github.com/mapbox/mapbox-gl-native/.
  - A library for embedding interactive, customizable vector maps into native applications on multi-platforms.
  - It takes stylesheets that conform to the Mapbox Style Specification, applies them to vector tiles that conform to the Mapbox Vector Tile Specification, and renders them using OpenGL.
  - Mapbox GL JS is a JavaScript library for interactive, customizable vector maps on the web.
  - It takes map styles that conform to the Mapbox Style Specification, applies them to vector tiles that conform to the Mapbox Vector Tile Specification, and renders them using WebGL.

Mapbox


E 20th St
E 14th St
E 13th St
E 11th St
ALPHABET CITY
Avenue D
FDR Dr
East River
East River Park
CALLOWHILL ST
WOOD ST
N 3RD ST
N BODINE ST
VINE ST
NEW ST
BENJAMIN FRANKLIN BRIDGE
FLORIST ST
RACE ST
N 2ND ST
Долина Каменушки
Южный лес
Sky Way
Liberty State Park
Pittsburgh
NOSC PITTSBURGH
IOWA AAP
CAPITAL AIRPORT
NOSC COLUMBUS
Cincinnati
St. Louis
ST LOUIS AFS
NSA CRANE
Washing
YEAGER AIRPORT
FORT LEON
RADFORD AAP
FORT CAMPBELL TN
Nashville
NOSC KNOXVILLE
Raleigh
FORT SMITH MAP
Memphis
NSWC CARDEROCK DIV
ARNOLD AS
Wilmington
Columbia
Atlanta
DOBBINS AFB
NWS CHARLESTON SH
Jackson
OLF BRAVO
MAXWELL AFB GUNTER ANNEX
HUNTER ARMY AIRFIELD
17R
18R
ENGLAND AFB
WHITING PINES
Jacksonville
NAS JACKSONVILLE - YELLOW WATER HOUSING
New Orleans
Tampa
NOTU CAPE CANAVERAL

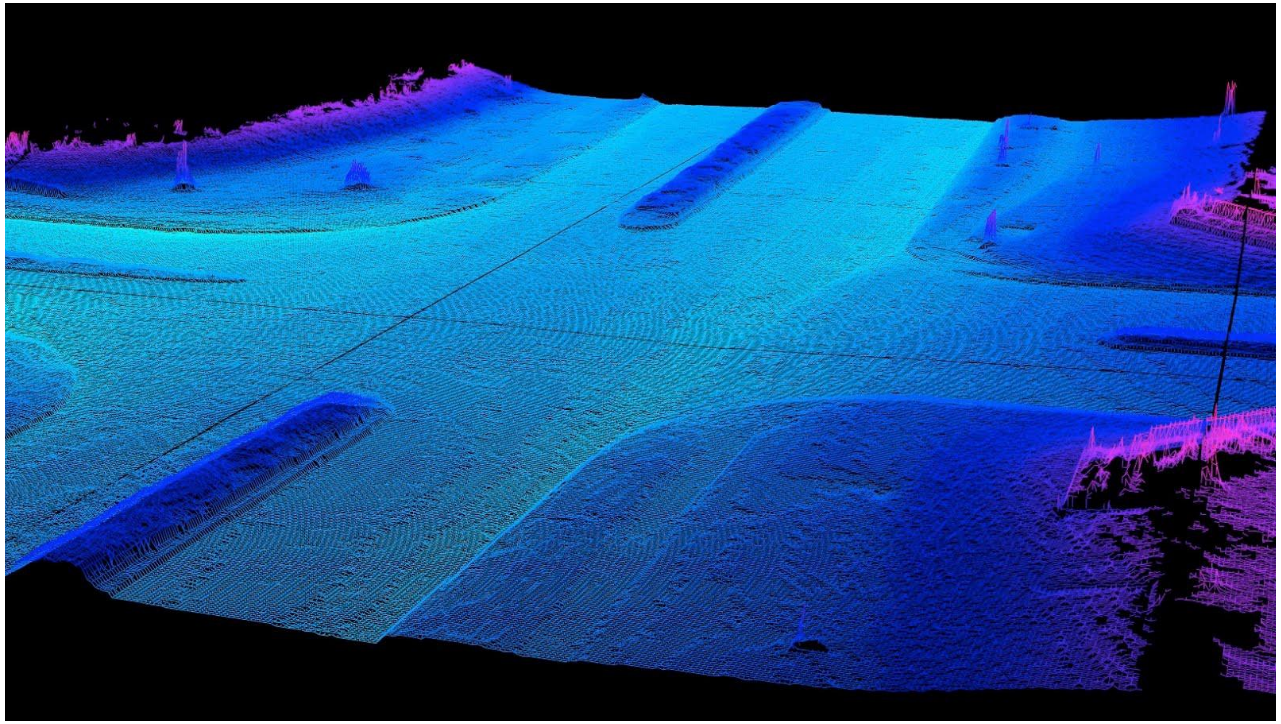
Simulation Platform at Waymo

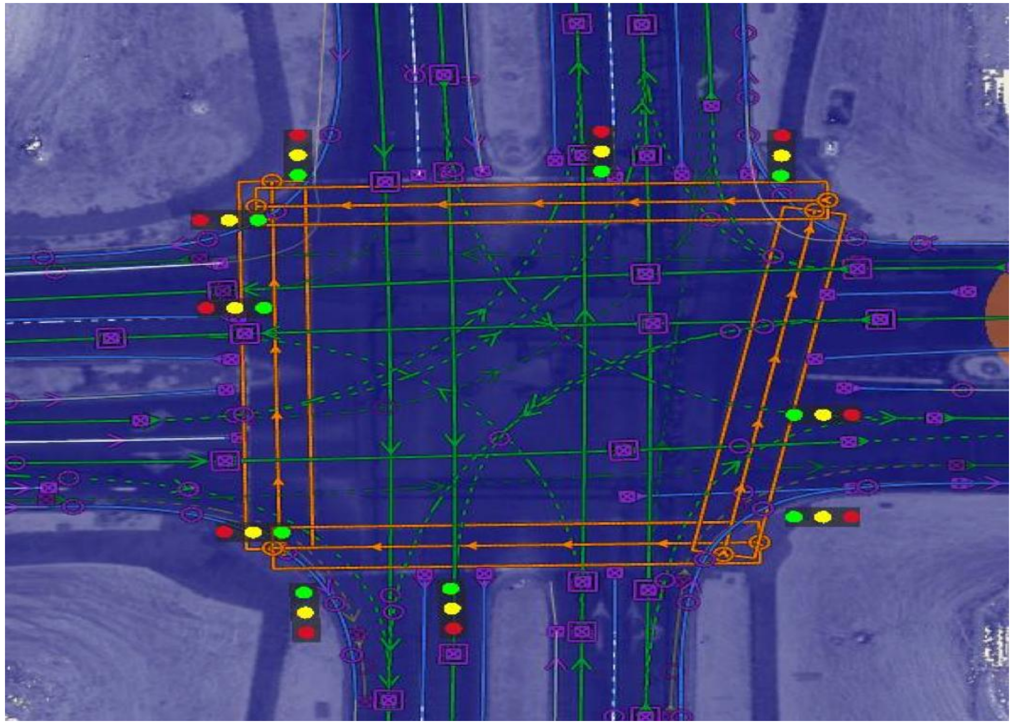
- Waymo’s simulator is a realistic virtual world to recreate every real-world mile they have driven.
    - Each day, as many as 25,000 virtual Waymo self-driving cars drive up to 8 million miles in simulation, testing out new skills and refining old ones.
- Like athletes visualizing the playing field, its virtual cars envision various scenarios and practice maneuvers that help them safely navigate the real world.



Simulation Platform at Waymo

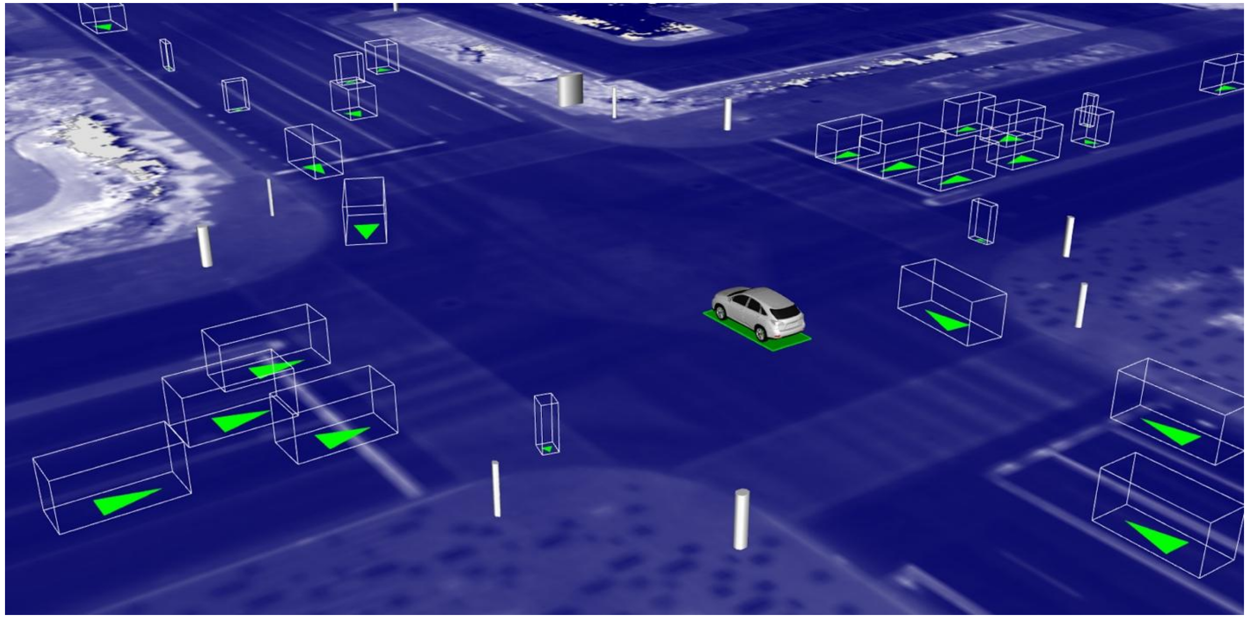
- A map for self-driving cars has a lot detail than conventional maps (e.g. the height of a curb, width of an intersection, and the exact location of a traffic light or stop sign);
- Lidars send out pulses of light that help painting a 3-d point cloud of the world.
- Categorize features on the road, such as driveways, fire hydrants, and intersections.





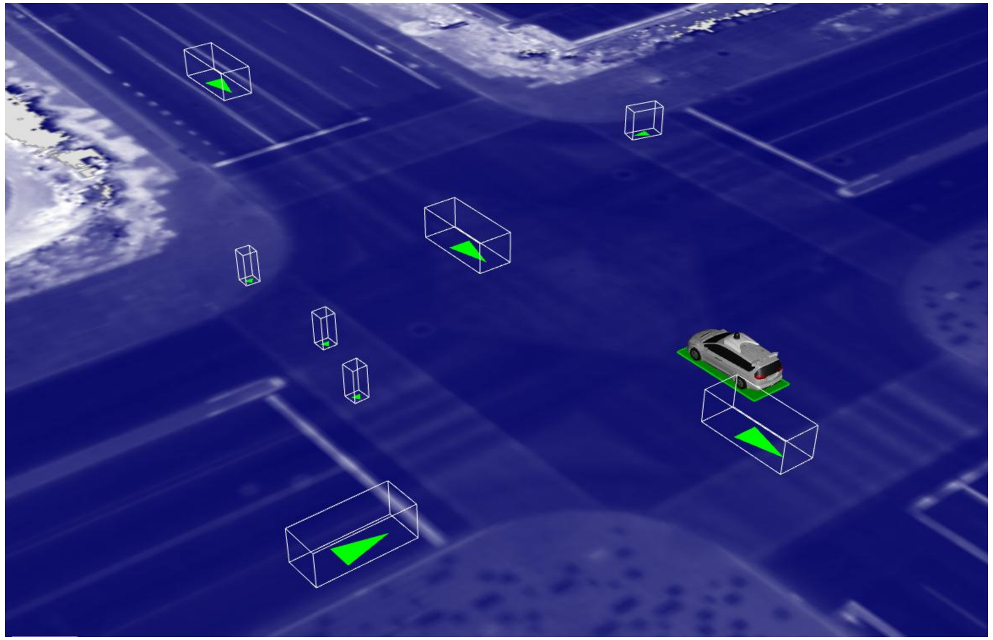
Simulation Platform at Waymo

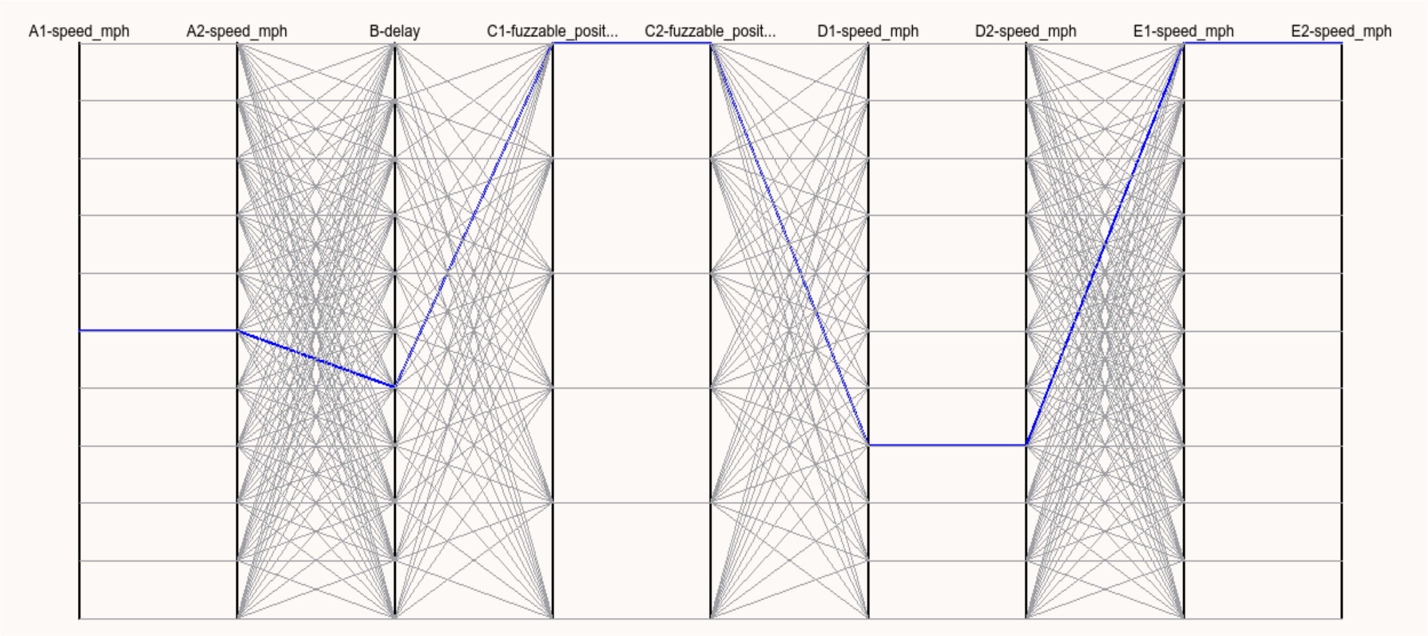
- One of the key advantages of simulation is that engineers and scientists can focus on the most interesting interactions, such as flashing yellow signals, wrong-way drivers, or nimble pedestrians and cyclists, rather than monotonous highway miles.
- With these scenarios in real world digitized in the virtual world, software can practice them thousands of times over, teach the cars how to handle them, and then practice what being learned to other places with the same scenarios.



Simulation Platform at Waymo

- Through a process called **fuzzing**, alter the speed of oncoming cars and the timing of traffic lights to make sure vehicles can still find the right gap in traffic to make a safe and smooth turn.
- Also make the street scene busier and more complex by adding vehicles, pedestrians and cyclists that never existed in the original scene.
- With fuzzing, even simulate motorcycles “splitting the lane,” or joggers zig-zagging across the street to see how that might change the driving on the road.

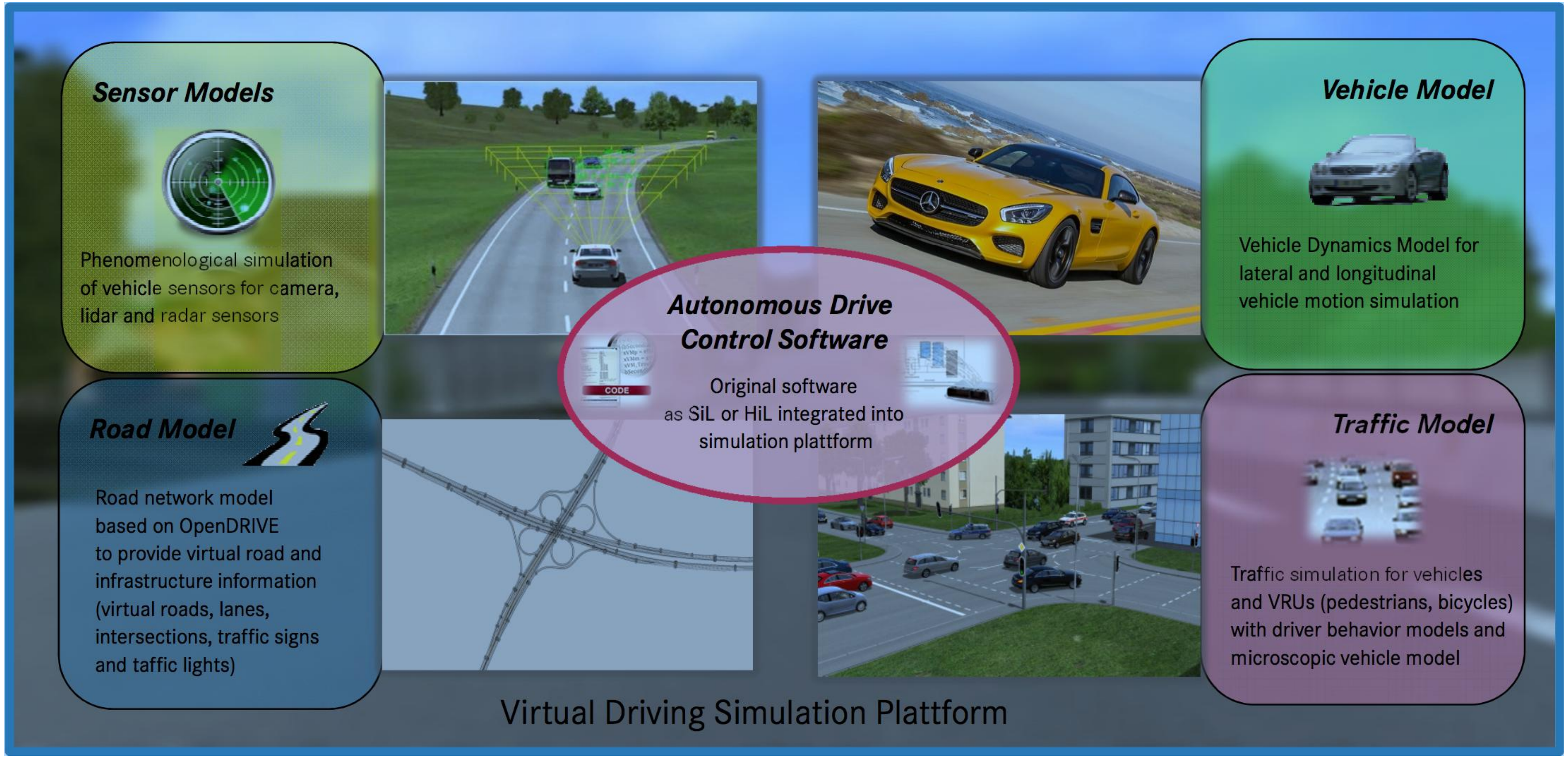




Simulation Platform at Waymo

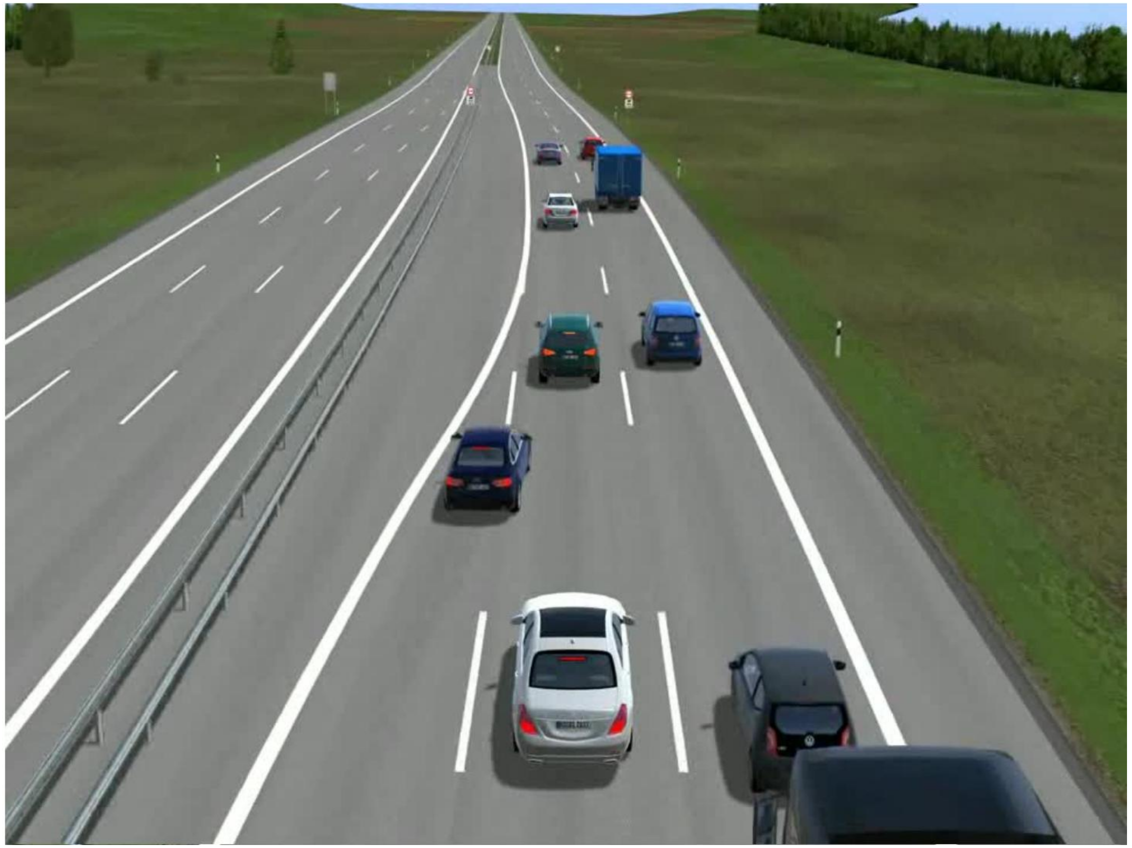
- Use real-world driving and private test track to verify and validate experience in simulation, and then the cycle begins again.
- Waymo cars drove over 2.5 billion simulated miles in 2016, miles far richer and more densely packed with interesting scenarios than the average mile of driving.
- Every mile useful to the goal of putting self-driving cars on the road, and saving lives.

Simulation Platform at Daimler



Simulation Platform at Daimler

- **Methods**:
  - Much more simulation, esp. for verification of control algorithms and rule compliance
  - Systematic search for rare functional deficits, instead of just driving test kilometers
- **Functions**
  - Continuous assessment of and adaptation to external conditions and rules
  - Judging reliably whether the limits of vehicle autonomy are close
  - Announce the end of autonomous mode early enough for the driver to take over (Level 3)
  - Bring the vehicle to a safe stop, if (in Level 3) the driver should fail to take over



SUMO (Simulation of Urban Mobility)

- It is mainly developed by **Institute of Transportation Systems at the German Aerospace Center**.
- Traffic simulations facilitate the evaluation of infrastructure changes as well as policy changes before implementing them on the road.
- It is a free open traffic simulation suite, licensed under the **GPL**, available since 2001.
- It allows modelling of intermodal traffic systems as road vehicles, public transport and pedestrians.
- Included with SUMO is a wealth of supporting tools which handle tasks such as route finding, visualization, network import and emission calculation.
- It can be enhanced with custom models and provides various APIs to remotely control simulation.

SUMO (Simulation of Urban Mobility)

- Simulation platform SUMO offers features:
  - Microscopic simulation - vehicles, pedestrians and public transport are modeled explicitly
  - Online interaction – control the simulation with TraCI
  - Simulation of multimodal traffic, e.g., vehicles, public transport and pedestrians
  - Time schedules of traffic lights can be imported or generated automatically by SUMO
  - No artificial limitations in network size and number of simulated vehicles
  - Supported import formats: OpenStreetMap, VISUM, VISSIM, NavTeq
  - SUMO is implemented in C++ and uses only portable libraries
- SUMO's projects for answering a large variety of research questions:
  - Evaluate perform. of traffic lights, evaluation of algorithms up to evaluation of weekly timing plans.
  - Vehicle route choice: development of new methods, evaluation of eco-aware routing based on pollutant emission, and investigations on network-wide influences of autonomous route choice.
  - To provide traffic forecasts for Cologne city during Pope's visit 2005 and during Soccer World Cup 2006.
  - To support simulated telephony behavior for evaluating performance of GSM-based traffic surveillance.
  - V2X community for realistic vehicle traces and evaluation in an on-line loop with a network simulator.

AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles

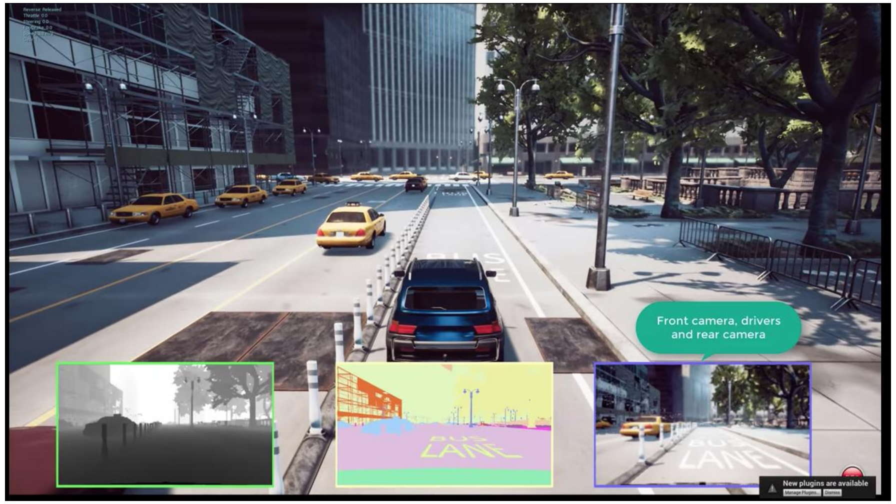
- Open source from MSR: https://github.com/Microsoft/AirSim;
- Developing and testing algorithms for autonomous vehicles;
- To collect a large amount of annotated training data in a variety of conditions and environments.
- A simulator built on Unreal Engine that offers physically and visually realistic simulations.
- A physics engine that can operate at a high frequency for real-time hardware-in-the-loop (HITL) simulations with support for popular protocols (e.g. MavLink).
- The simulator is designed from the ground up to be extensible to accommodate new types of vehicles, hardware platforms and software protocols.

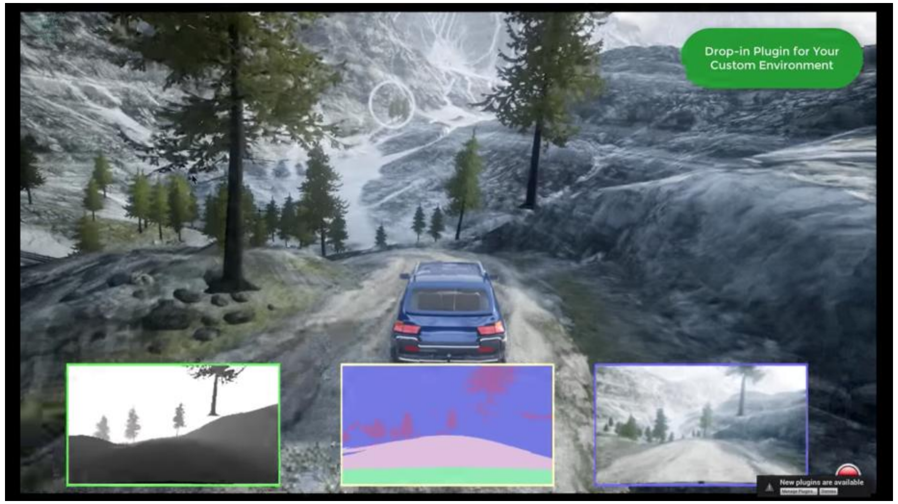
AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles

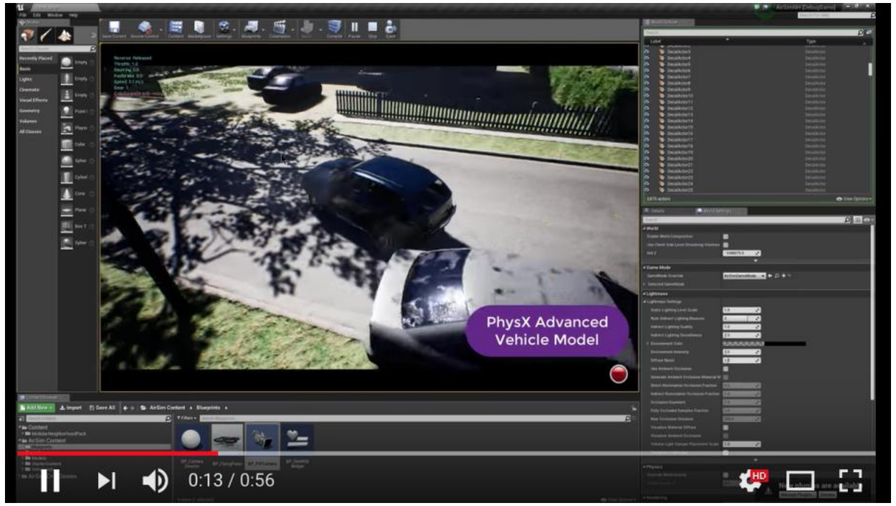
A snapshot from AirSim shows an aerial vehicle flying in an urban environment. The inset shows depth, object segmentation and camera streams generated in real time.

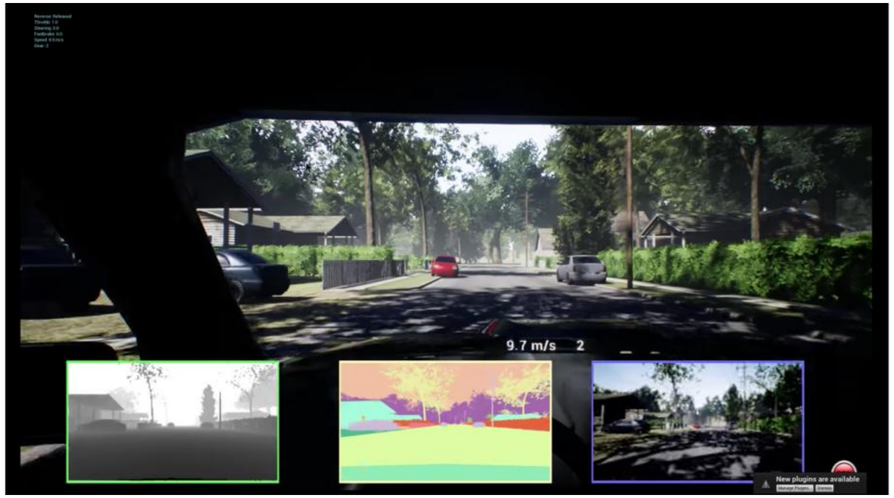
AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles

Cars in AirSim

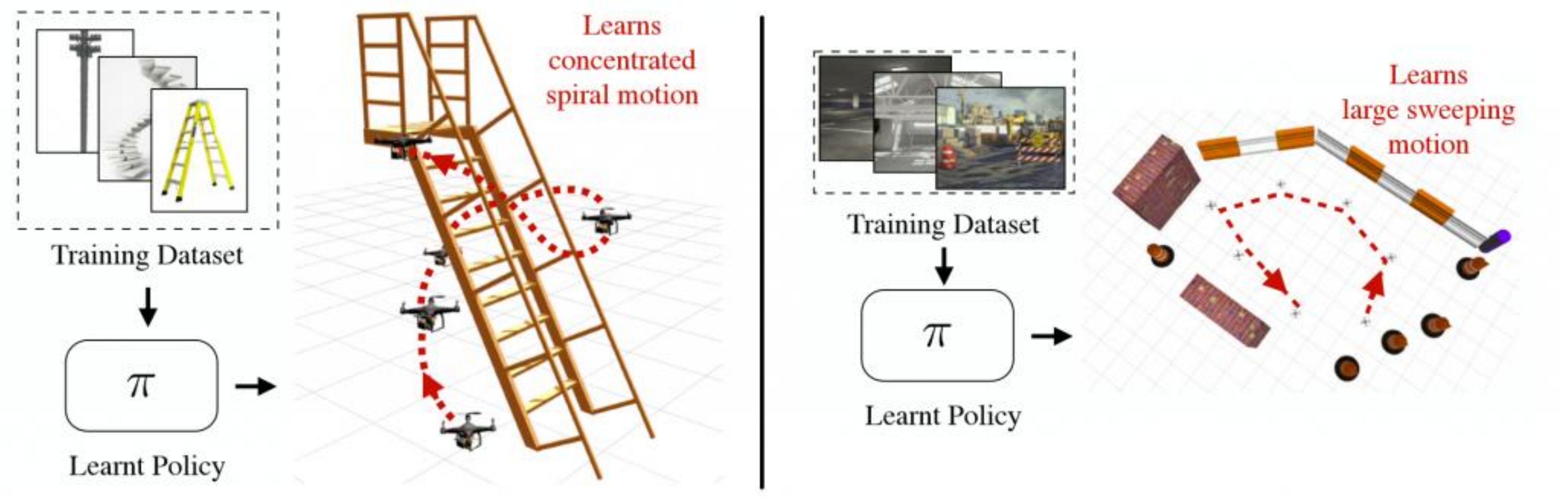






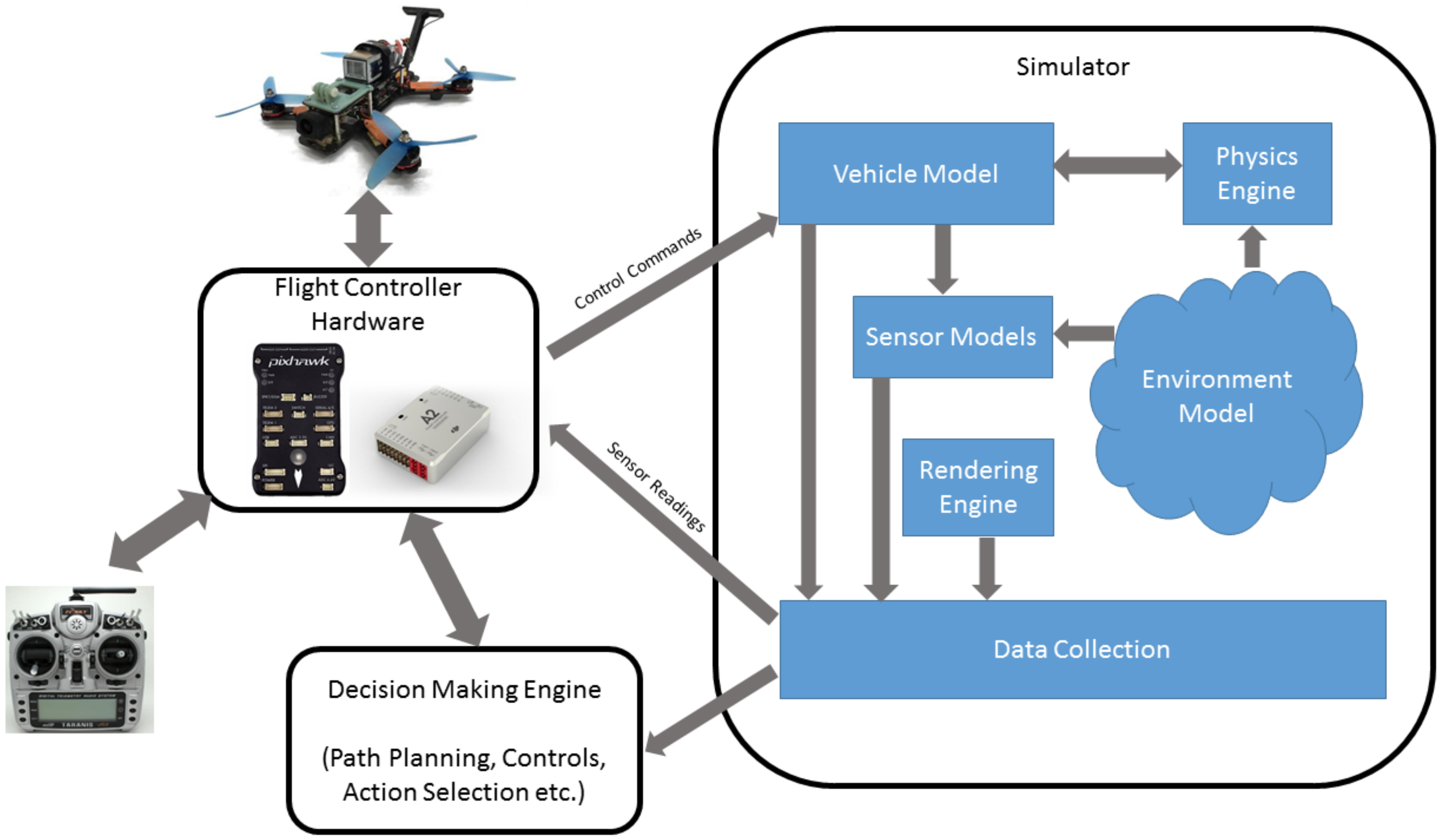


AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles



This figure shows the ability to generalize to different structured environments. The flying quadrotor, using the same underlying mechanism, learns to avoid obstacles autonomously for different environments.

AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles



The architecture of the simulation system that depicts the core components and their interactions.

CARLA: An Open Urban Driving Simulator

- CARLA as open source, developed to support development, training, and validation of autonomous urban driving systems.
- CARLA provides open digital assets (urban layouts as street signs, buildings, vehicles);
- CARLA supports setup of sensor suites and provides signals used to train driving strategies, such as GPS coordinates, speed, acceleration, and detailed data on collisions and other infractions.
- A wide range of environmental conditions can be specified, including weather and time of day;
- Three approaches to autonomous driving in CARLA:
  - A classic modular pipeline;
  - An end-to-end model trained via imitation learning;
  - An end-to-end model trained via reinforcement learning.

CARLA: An Open Urban Driving Simulator

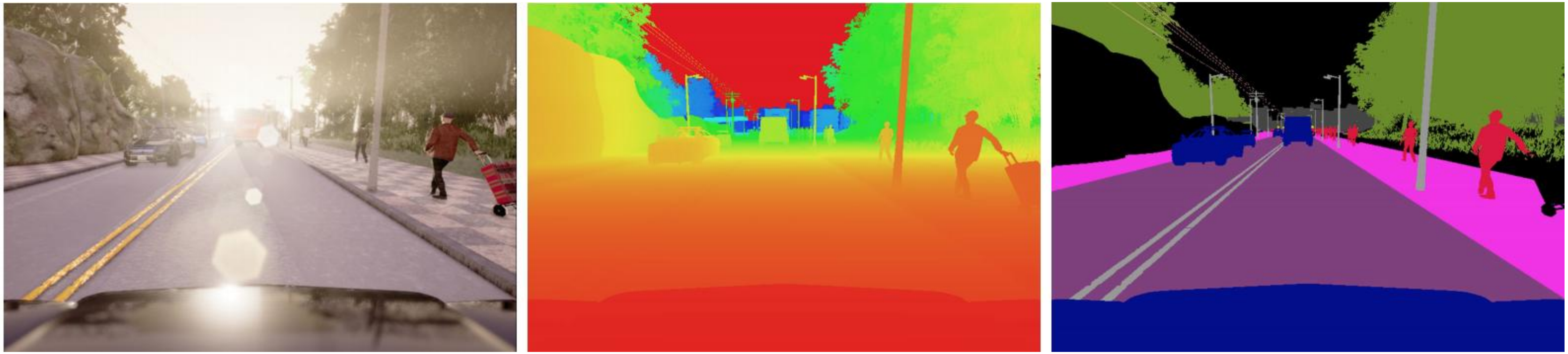
- Simulation can democratize research in autonomous urban driving, which is also necessary for system verification, since some scenarios are too dangerous to be staged in the physical world.
- Commercial games have been used to acquire high-fidelity data for training and benchmarking visual perception systems.
- Simulator TORCS do not present complexity of urban driving: they lack pedestrians, intersections, cross traffic, traffic rules, and other complications that distinguish urban driving from track racing.
- Commercial game Grand Theft Auto V, do not support detailed benchmarking of driving policies:
    - little customization and control over environ.
    - limited scripting and scenario specification
    - severely limited sensor suite specification
    - no detailed feedback upon violation of rules
    - limitations due to closed-source commercial nature and different objectives in developing.

CARLA: An Open Urban Driving Simulator

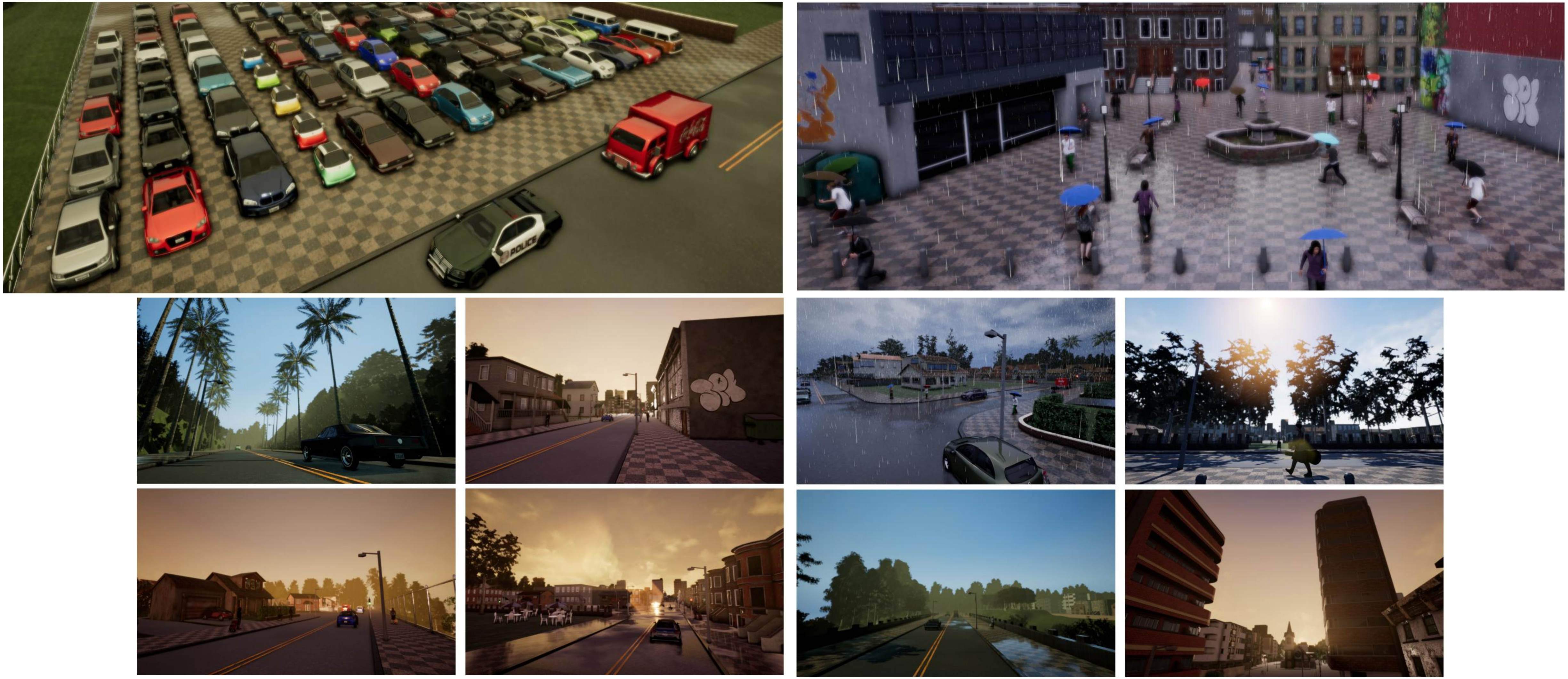
- CARLA implemented as an open-source over Unreal Engine 4 (UE4) enabling future extensions.
- The engine provides state-of-the-art rendering quality, realistic physics, basic NPC logic, and an ecosystem of interoperable plugins.
- CARLA is designed as a server-client system, where the server runs the simulation and renders the scene, while the client API is implemented in Python for the interaction btw the autonomous agent and the server via sockets.
- The client sends commands and meta-commands to the server and receives sensor readings in return, commands for steering, accelerating, and braking, meta-commands for resetting the simulation, changing the properties of the environment, and modifying the sensor suite.
- Steps to build urban environments: (a) laying out roads and sidewalks; (b) manually placing houses, vegetation, terrain, and traffic infrastructure; and (c) specifying locations where dynamic objects can appear (spawn).
- A basic controller for non-player vehicle behavior: lane following, respecting traffic lights, speed limits, and decision making at intersections.

CARLA: An Open Urban Driving Simulator

- Sensors are limited to RGB cameras and to pseudo-sensors that provide ground-truth depth and semantic segmentation;
- 12 semantic classes: road, lane-marking, traffic sign, sidewalk, fence, pole, wall, building, vegetation, vehicle, pedestrian, and other;
- Measurements of the agent’s state include vehicle location and orientation wrt the world coordinate system (akin to GPS and compass), speed, acceleration vector, and accumulated impact from collisions.



CARLA: An Open Urban Driving Simulator

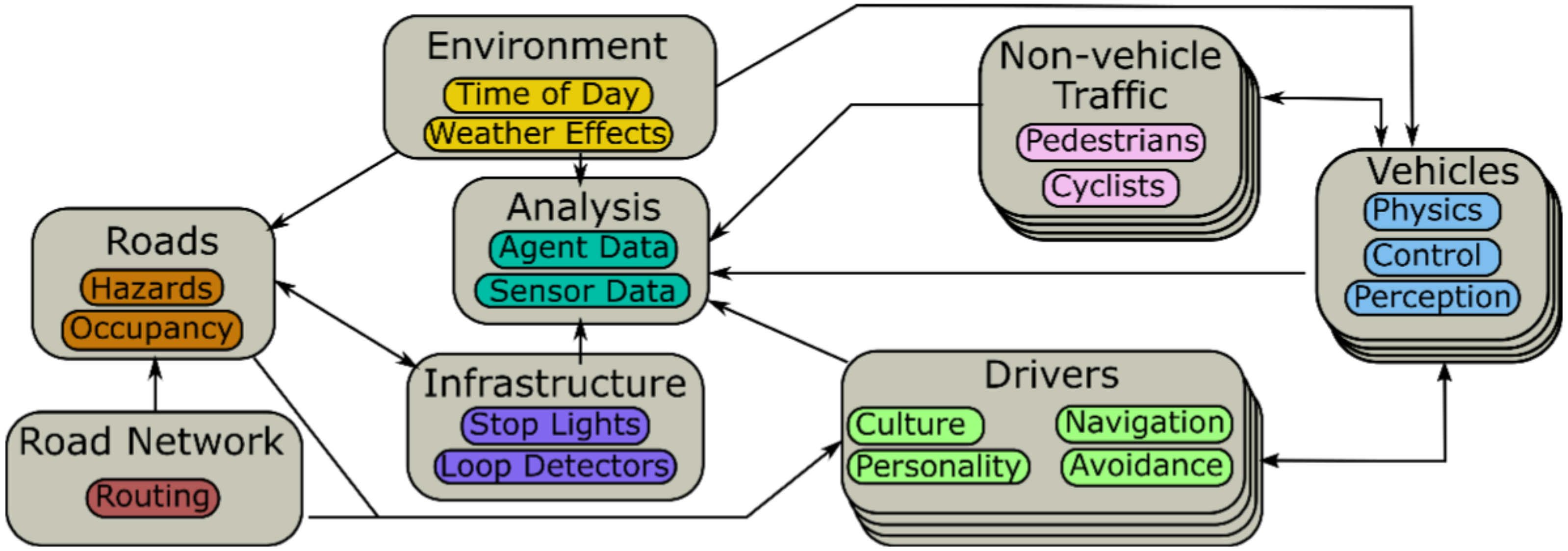


AutonoVi-Sim: Autonomous Vehicle Simulation

- It is a simulation platform for autonomous driving data generation and driving strategy testing.
- A collection of high-level extensible modules which allows the rapid development and testing of vehicle configurations and facilitates construction of complex traffic scenarios.
- It supports multiple vehicles with unique steering or acceleration limits, as well as unique tire parameters and dynamics profiles.
- Engineers can specify vehicle sensor systems and vary time of day and weather conditions and gain insight into how conditions affect the performance.
- It supports navigation for non-vehicle traffic participants such as cyclists and pedestrians, allowing engineers to specify routes for these actors, or to create scripted scenarios which place the vehicle in dangerous reactive situations.
- It facilitates training of deep-learning algorithms by enabling data export from the vehicle’s sensors, including camera, LIDAR, relative positions of traffic participants, and detection/classification results.
- It allows for the rapid prototyping, development and testing of autonomous driving algorithms under varying vehicle, road, traffic, and weather conditions.

AutonoVi-Sim: Autonomous Vehicle Simulation

- Insights gained from simulation could provide critical training data and information on algorithmic inefficiencies before actual vehicle testing.
- AutonoVi-Sim is divided into 8 extensible modules, each with various sub-components.
  - Environment, Road Network, Road, Drivers, Infrastructure, Vehicles, Non-vehicle Traffic, and Analysis.
  - Each module captures some aspect of autonomous driving simulation, extended/modified to suit the needs.



AutonoVi-Sim: Autonomous Vehicle Simulation

- The Road, Road Network, and Infrastructure modules define the driving environment.
- The Environment module allows engineers to specify specific environment conditions including time of day and weather.
- The Non-vehicle Traffic module allows engineers to specify navigation goals for pedestrians and cyclists, or setup specific triggered behaviors.
- The Drivers and Vehicles modules work as a pair to define current traffic conditions and specific driving destinations and decisions for the vehicles in the simulation.
- Each vehicle in the simulation has a unique set of sensing capabilities and a single driver which operates the vehicle during the simulation.
- The Analysis module is used to catalog and export data, including agent positions and sensor readings, for analysis.

AutonoVi-Sim: Autonomous Vehicle Simulation

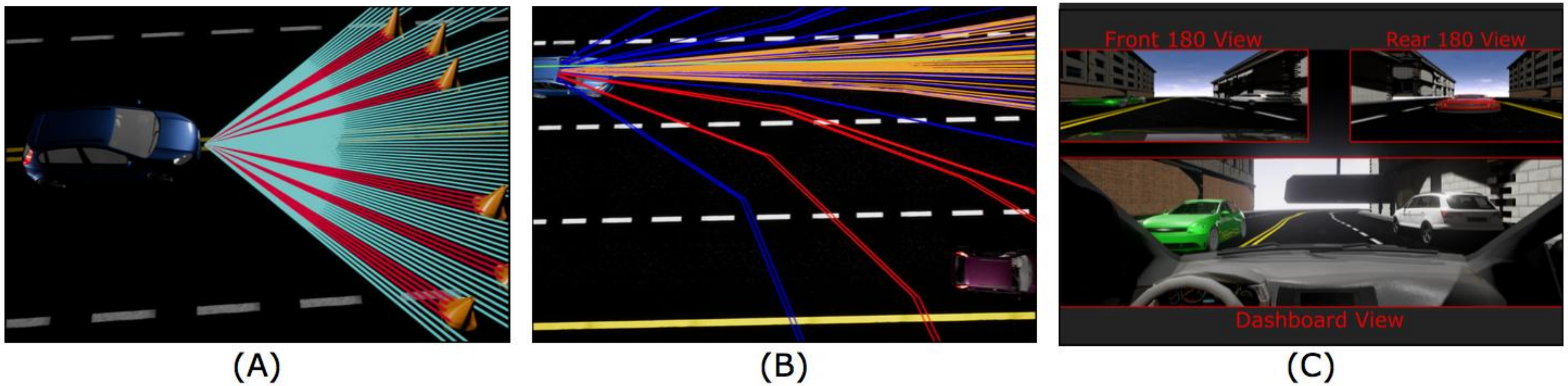


Figure 3: **AutonoVi-Sim Modules:** **(A):** Sensors on the vehicle are placed interactively. This configuration demonstrates a hatchback with a laser rangefinder navigating around traffic cones. Returned beams are illustrated in red. Beams which do not return data are illustrated in cyan for debugging. **(B)**: Once sensors are placed, the vehicle's navigation algorithm can be tested and examined interactively. The AutonoVi driving algorithm samples potential controls and projects forward in time. Red control paths indicate predicted collisions with the nearby vehicle. **(C)**: The data analysis module allows for exporting sensor data as the vehicle navigation. This test vehicle is equipped with a 180 degree forward facing camera, a 180 degree rear-facing camera, and a dashboard camera.

AutonoVi-Sim: Autonomous Vehicle Simulation

(A) (B) (C)

Figure 4: **Simulated scenarios and conditions in Autonovi-Sim: (A)**: A simulated city modelled in AutonoVi-Sim. Closed circuit road networks allow engineers to test driving algorithms over long timescales by assigning new navigation goals periodically. **(B)**: Heavy fog obstructs the view of the vehicle. **(C)**: Vehicles pass through a slick intersection during rainy conditions.

CityScapes DataSet

- The Cityscapes Dataset focuses on semantic understanding of urban street scenes;
- Type of annotations:
  - Semantic, Instance-wise, Dense pixel annotations
- Complexity: 30 classes
- Diversity:
  - 50 cities, Several months (spring, summer, fall), Daytime, Good/medium weather conditions, Manually selected frames
- Volume:
  - 5 000 annotated images with fine annotations, 20 000 annotated images with coarse annotations
- Metadata:
  - Preceding and trailing video frames, Each annotated image is the 20th image from a 30 frame video snippets (1.8s), Corresponding right stereo views, GPS coordinates, Ego-motion data from vehicle odometry, Outside temperature from vehicle sensor
- Benchmark suite and evaluation server:
  - Pixel-level semantic labeling, Instance-level semantic labeling

CityScapes DataSet

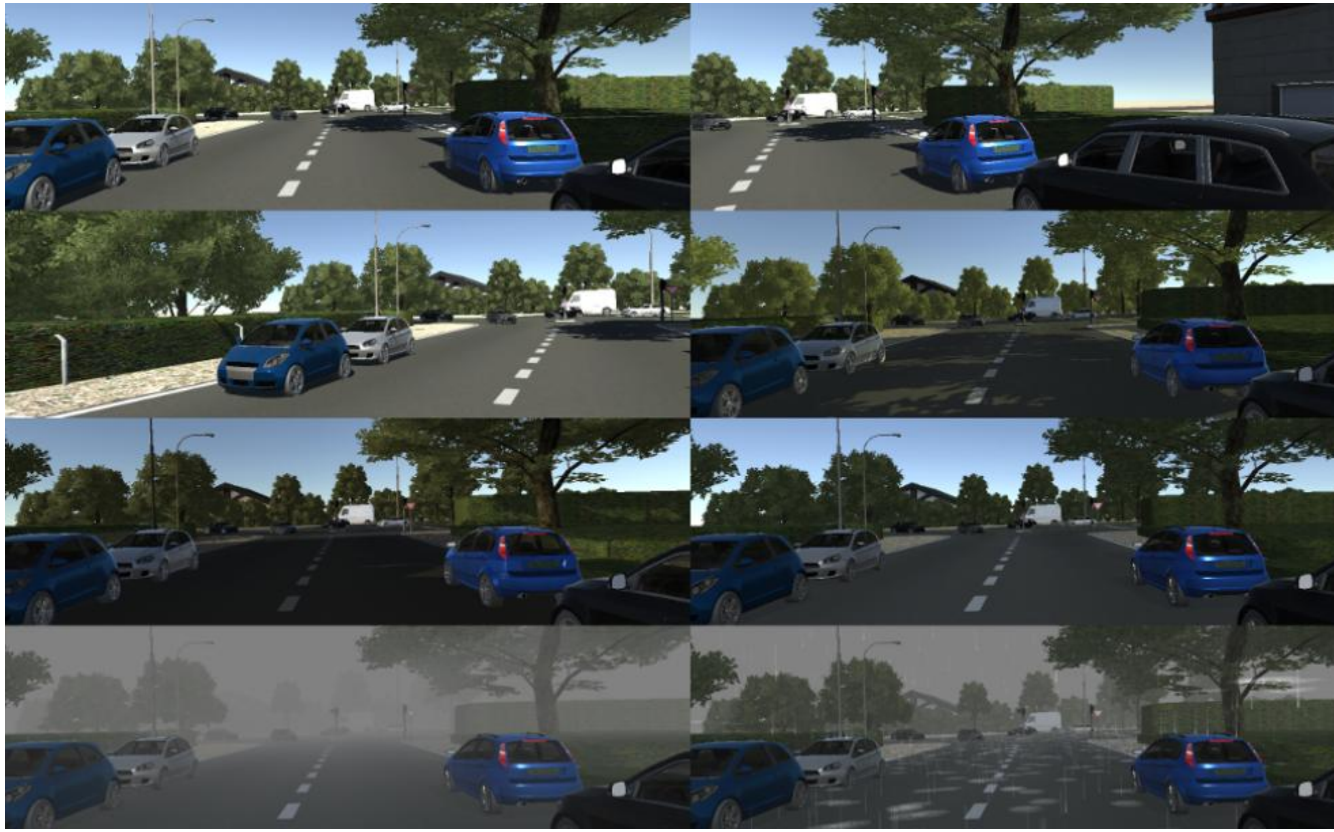
fine annotations

coarse annotations

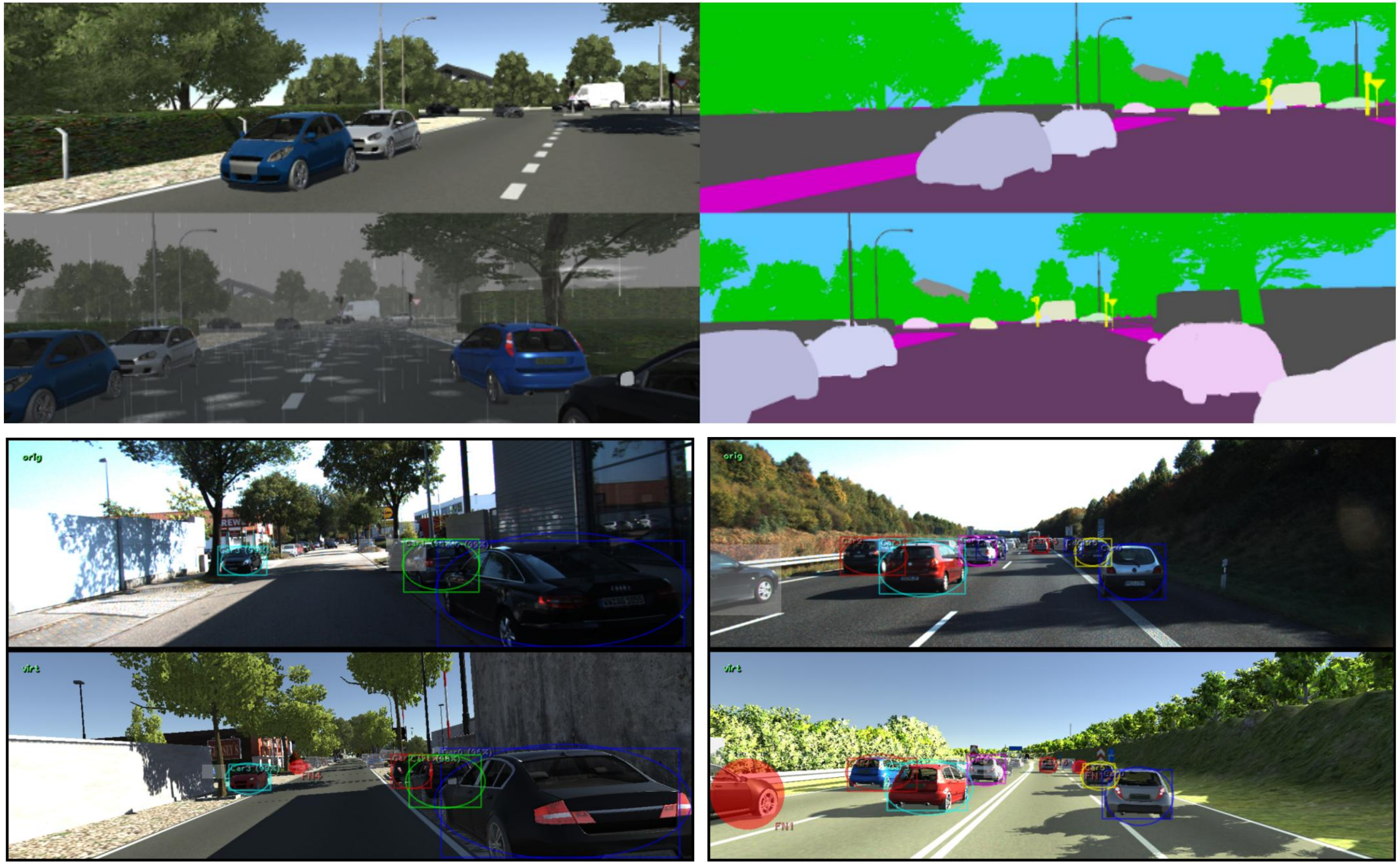
Virtual KITTI Dataset

- Virtual KITTI is a **photo-realistic synthetic video dataset** designed to learn and evaluate computer vision models for several **video understanding** tasks: object detection and multi-object tracking, scene-level and instance-level semantic segmentation, optical flow, and depth estimation.
- Virtual KITTI contains **50 HR monocular videos** (21,260 frames) generated from **five different virtual worlds in urban settings under different imaging and weather conditions**.
- These were created using Unity engine and a real-to-virtual cloning method.
- These photo-realistic synthetic videos are **automatically, exactly, and fully annotated for 2D and 3D multi-object tracking and at the pixel level with category, instance, flow, and depth labels**.

Virtual KITTI Dataset



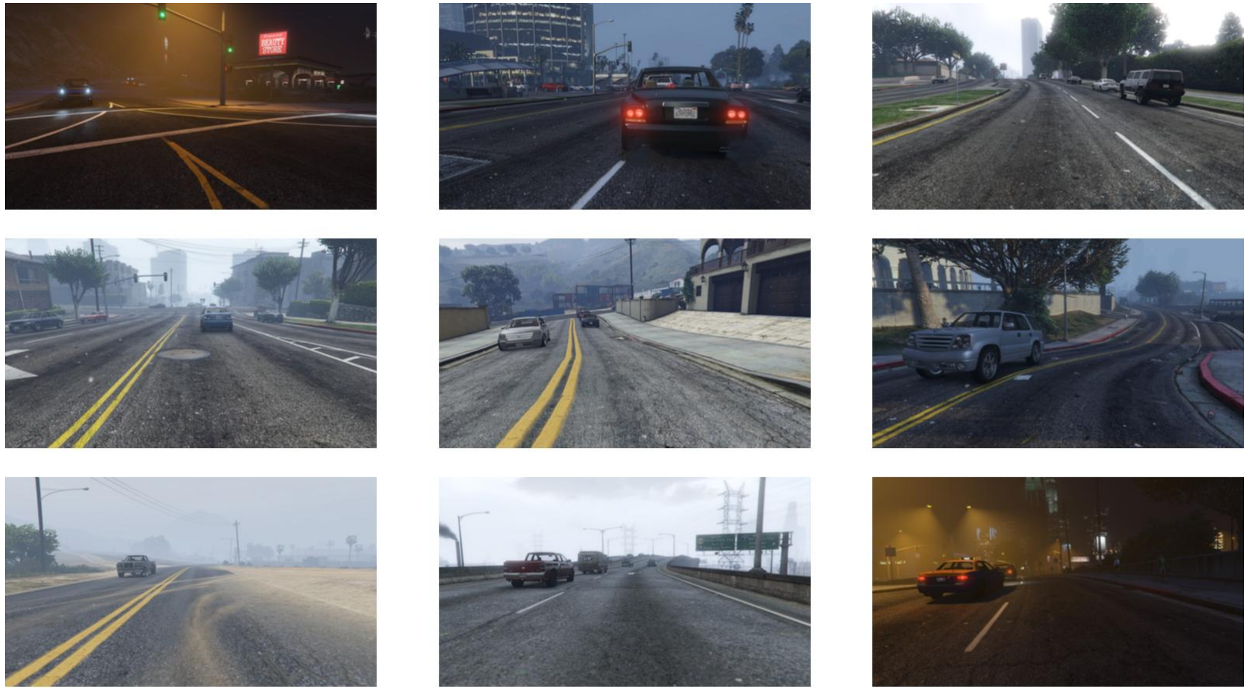
Virtual Kitti Dataset



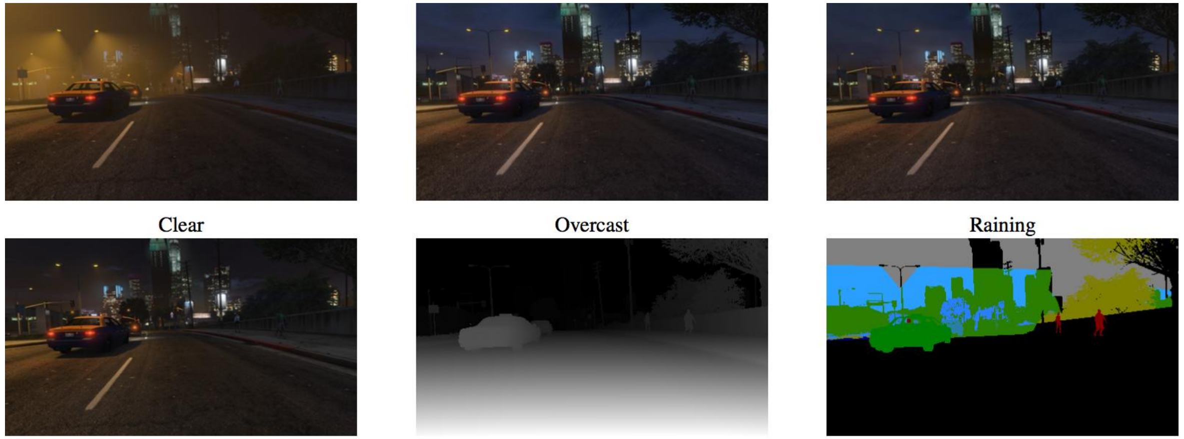
Simulation at FCAV (UM Ford Center for Autonomous Vehicles)

- Open source: https://github.com/umautobots/GTAVisionExport;
- Public dataset: https://fcav.engin.umich.edu/sim-dataset;
- Training: https://github.com/umautobots/driving-in-the-matrix;
- Rich virtual worlds created for major video games to simulate the real world with a high level of fidelity: Grand Theft Auto V (GTA V);
- Data is captured from the game using plugins: Script Hook V(.NET);
  - Sim 10k with 10,000 images, Sim 50k with 50, 000 images, and Sim 200k with 200, 000 images.

Simulation at FCAV (UM Ford Center for Autonomous Vehicles)



different times of day are simulated including day, night, morning and dusk, as well as complex weather and lighting scenarios such as driving into the sun, fog, rain and haze.

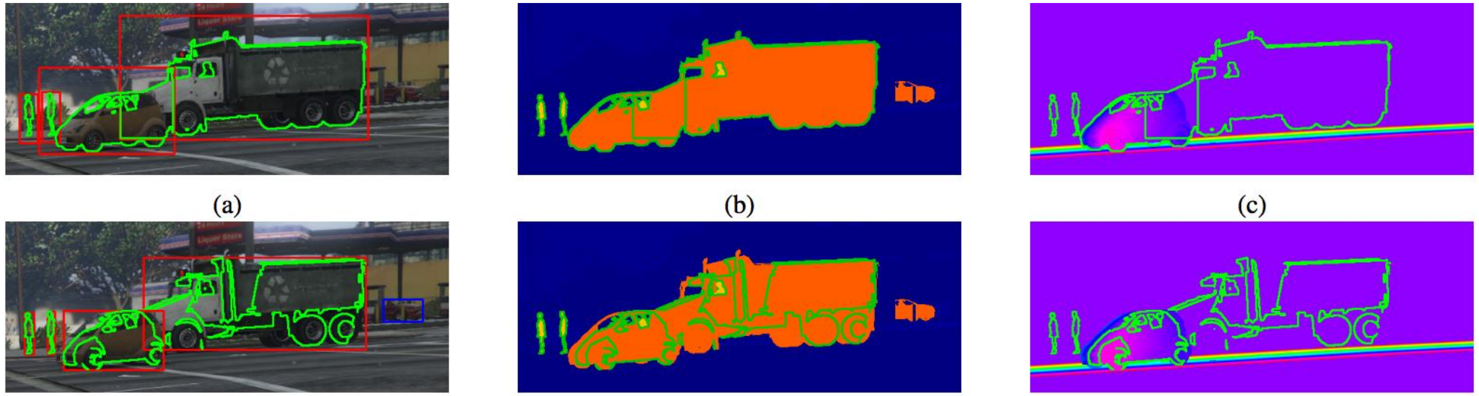


Clear | Overcast | Raining

Light Snow | Scene Depth | Pixel-wise Object Stencil Buffer

different weather types appear in the training data

Simulation at FCAV (UM Ford Center for Autonomous Vehicles)



(a) (b) (c)

(d) (e) (f)

pipeline for tight bounding box creation. original bounding boxes in (a). In (b) the orange pixels have been marked as vehicle to produce tight contours outlined in green. depth in (c) where lighter purple indicates closer range. (e) contains updated contours after processing using depth and (f) contains those same updated contours in the depth frame. (d) depicts the bounding boxes with the additional small vehicle detections in blue.

(a) 10k

(b) 50k

(c) 200k

increasing volumes of simulation data

Synthia Dataset

- SYNTHIA: SYNTHetic collection of Imagery and Annotations;
- Synthetic Images for Semantic Segmentation of Urban Scenes in the context of driving scenarios;
- It consists of a collection of photo-realistic frames rendered from a virtual city and comes with precise pixel-level semantic annotations for 13 classes: misc, sky, building, road, sidewalk, fence, vegetation, pole, car, sign, pedestrian, cyclist, lanemarking.
- +200,000 HD images from video streams and +20,000 HD images from independent snapshots;
- Scene diversity: European style town, modern city, highway and green areas;
- Variety of dynamic objects: cars, pedestrians and cyclists;
- Multiple seasons: dedicated themes for winter, fall, spring and summer;
- Lighting conditions and weather: dynamic lights and shadows, several day-time modes, rain mode and night mode;
- Sensor simulation: 8 RGB cameras forming a binocular 360º camera, 8 depth sensors;
- Automatic ground truth: individual instances for semantic segmentation (pixelwise annotations), depth, ego-motion.

Synthia Dataset

Figure 2. Dynamic objects catalogue of SYNTHIA. (Top) vehicles examples; (middle) cyclists; (bottom) pedestrians.

Synthia Dataset

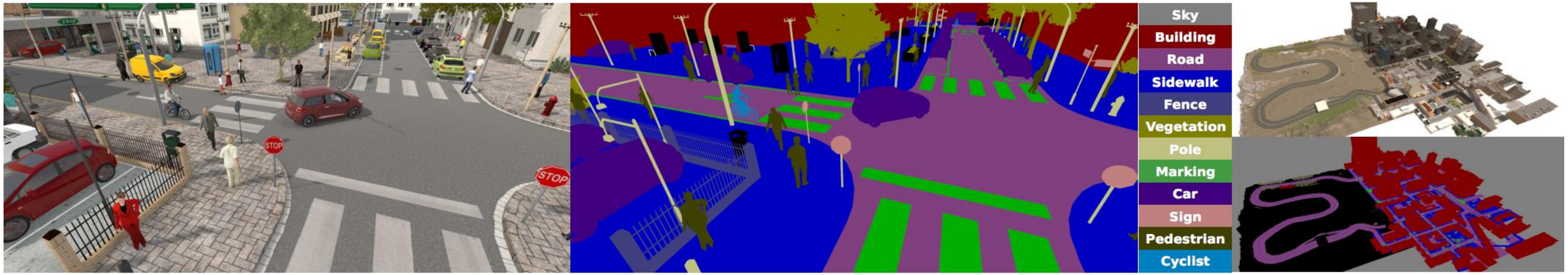


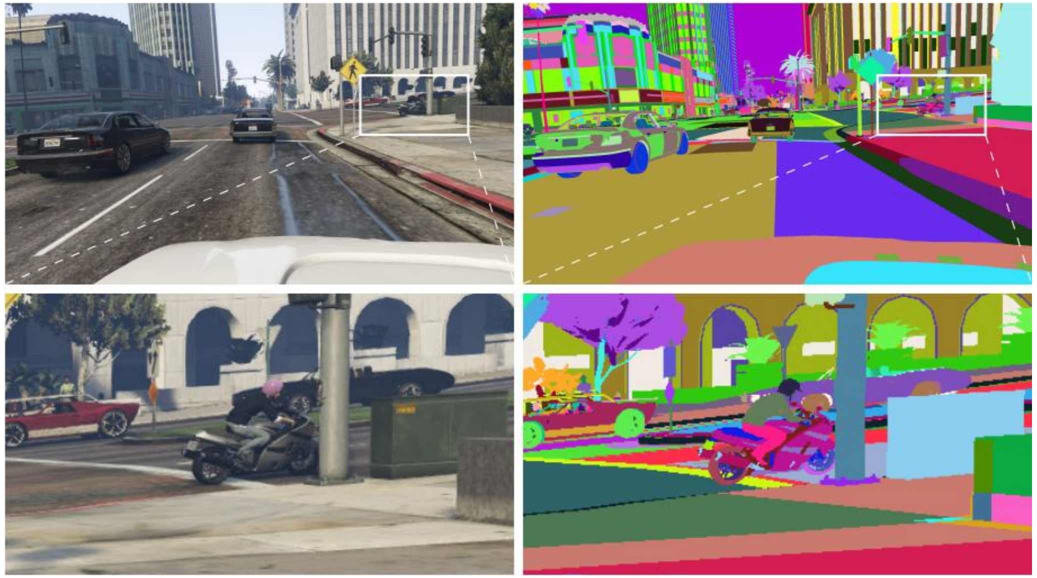
Figure. 1. The SYNTHIA Dataset. A sample frame (Left) with its semantic labels (center) and a general view of the city (right).

Figure 5. One shot example: the four views from the left multi-camera with its associated depth maps.

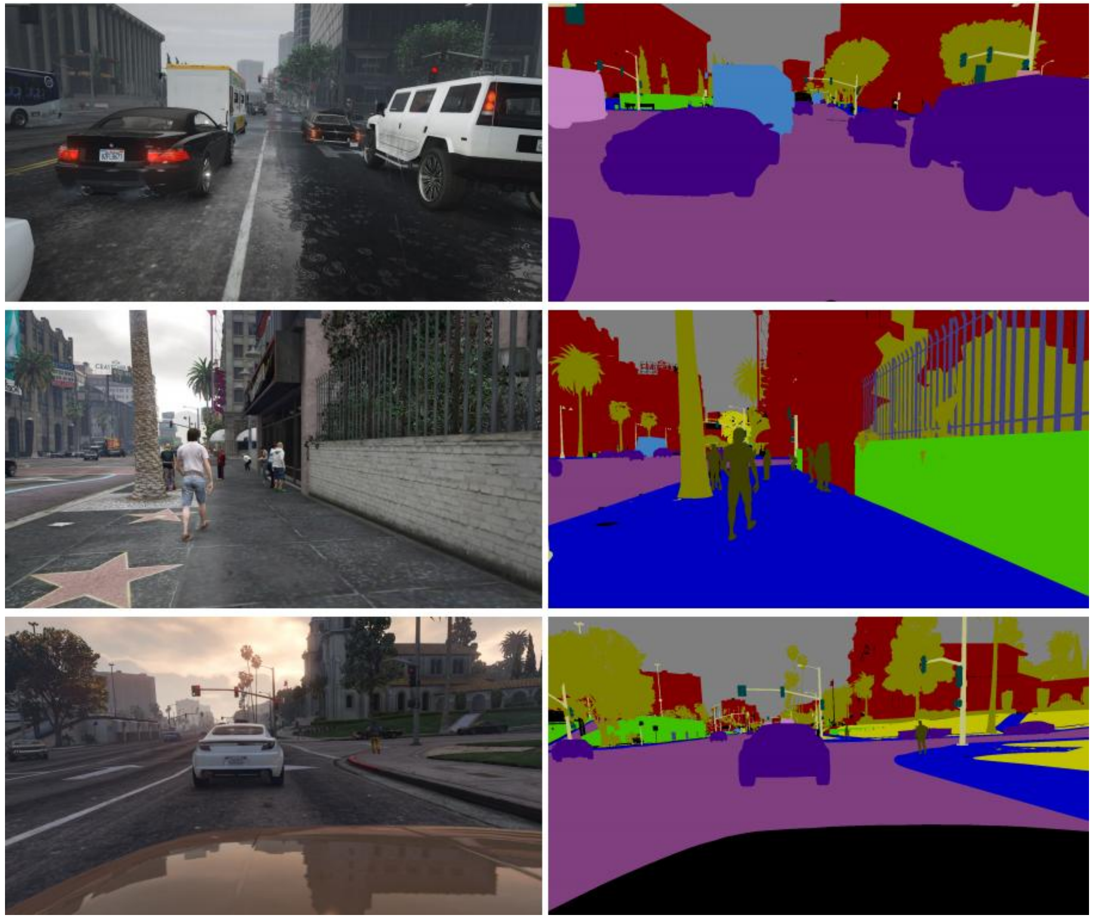
Semantic Segmentation Ground Truth from Computer Games

- Rapidly creating pixel-accurate semantic label maps for images extracted from modern computer games;
- Associations btw image patches can be reconstructed from the communication between the game and the graphics hardware;
- Rapid propagation of semantic labels across images synthesized by the game, with no access to the source code or the content;
- 25k images synthesized by a photorealistic open-world computer game: Grand Theft Auto V.
- Link: https://download.visinf.tu-darmstadt.de/data/from_games/.

Semantic Segmentation Ground Truth from Computer Games

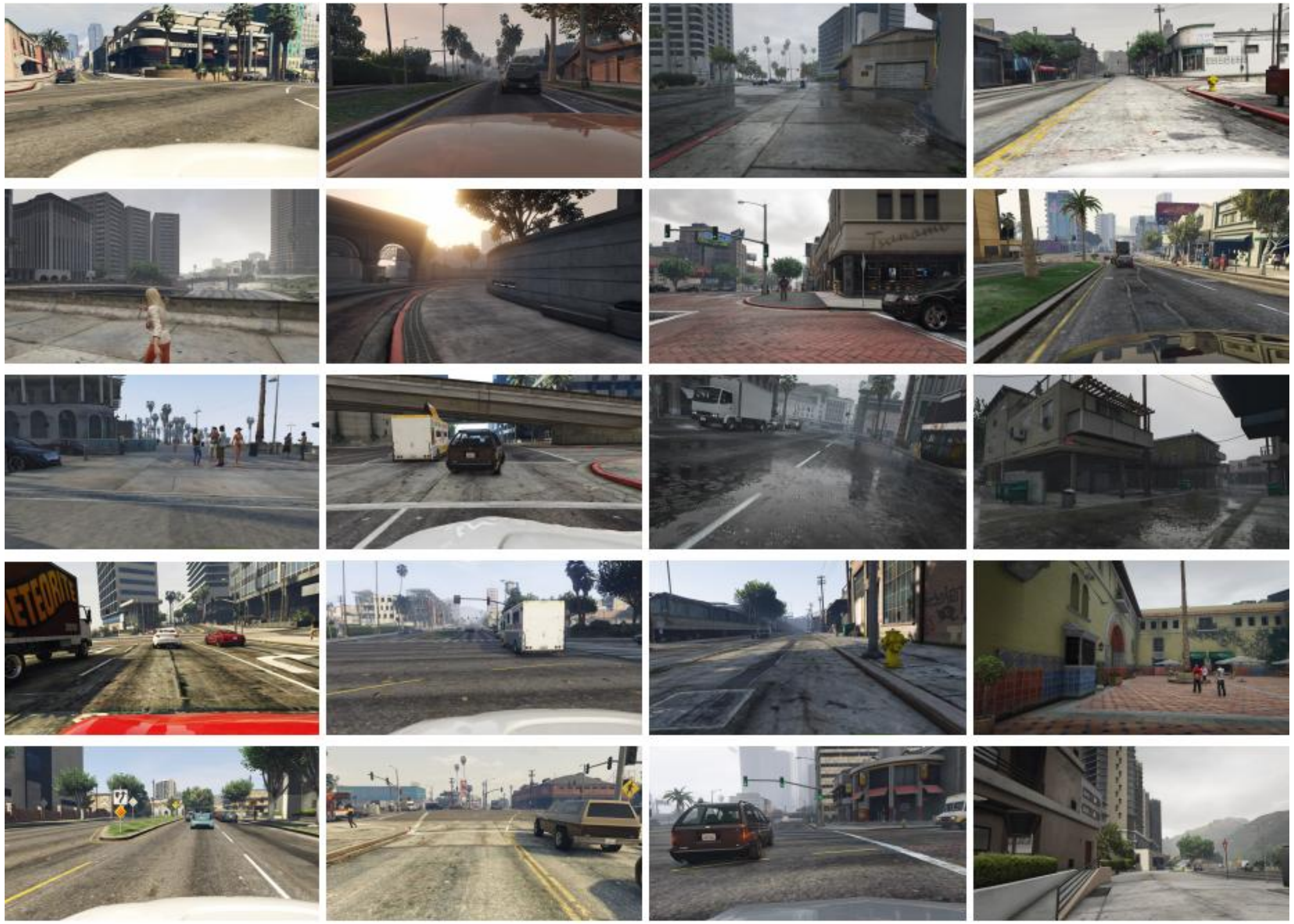


Automatically decomposing each image into patches of pixels that share a common <mesh, texture, shader> combination (henceforth, MTS).



Left: images extracted from the game Grand Theft Auto V. Right: semantic label maps.

Semantic Segmentation Ground Truth from Computer Games

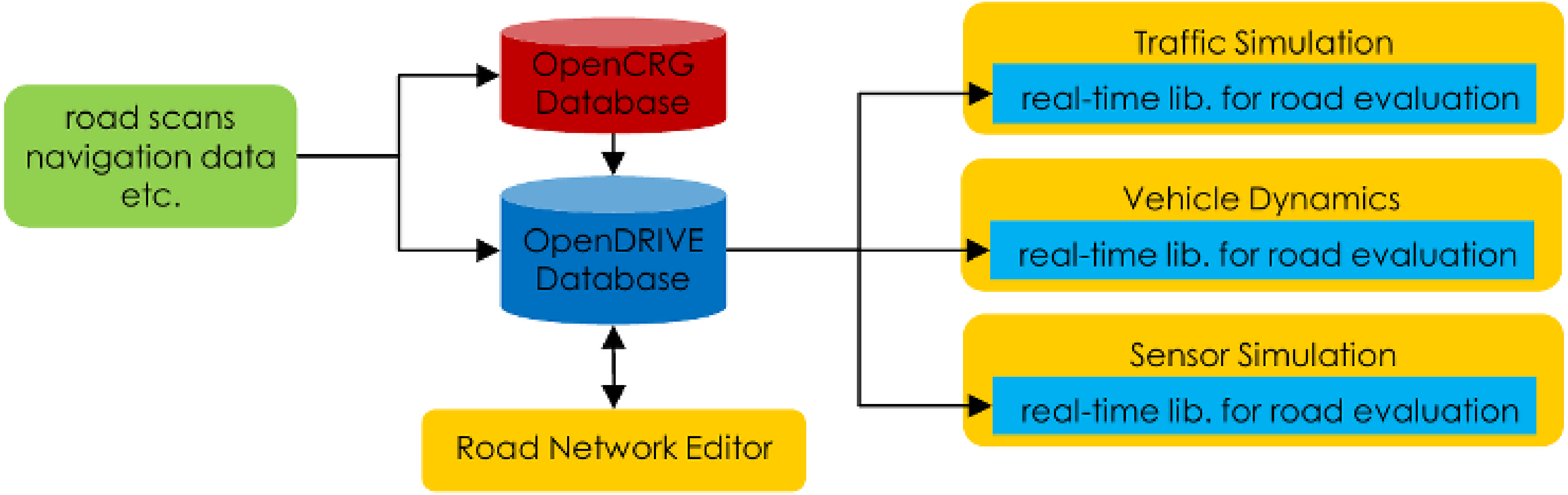


OpenDRIVE

- OpenDRIVE® is an **open file format** for the logical description of road networks, since Jan, 2006.
- It was developed and is being maintained by a team of simulation professionals with large support from the simulation industry.
- In order to interface the databases with the vehicle dynamics or the autonomous traffic, an additional description of a road network's logics had to be delivered with each visual database.

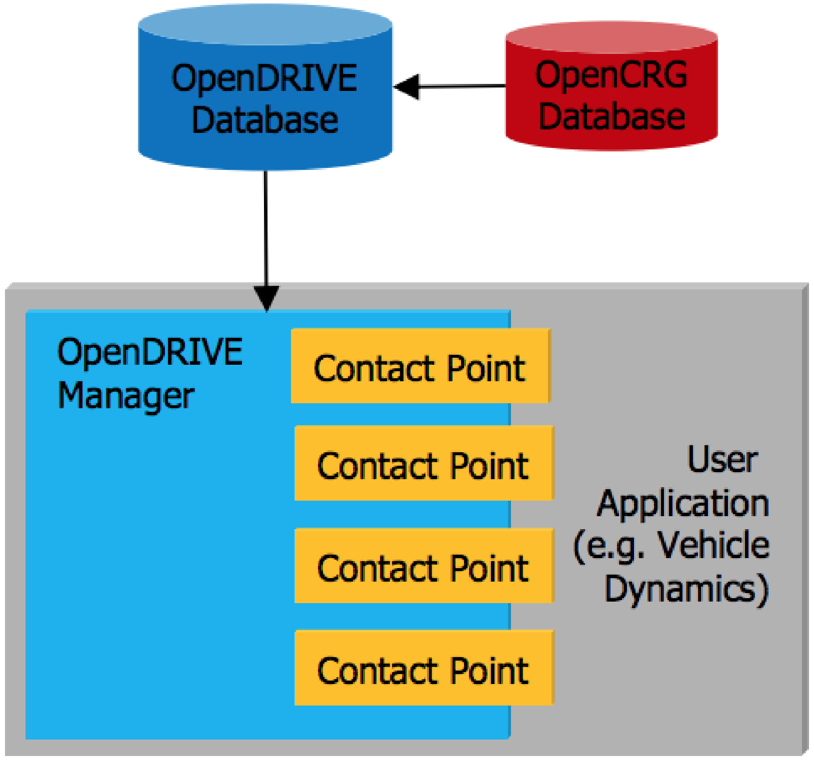
OpenDRIVE

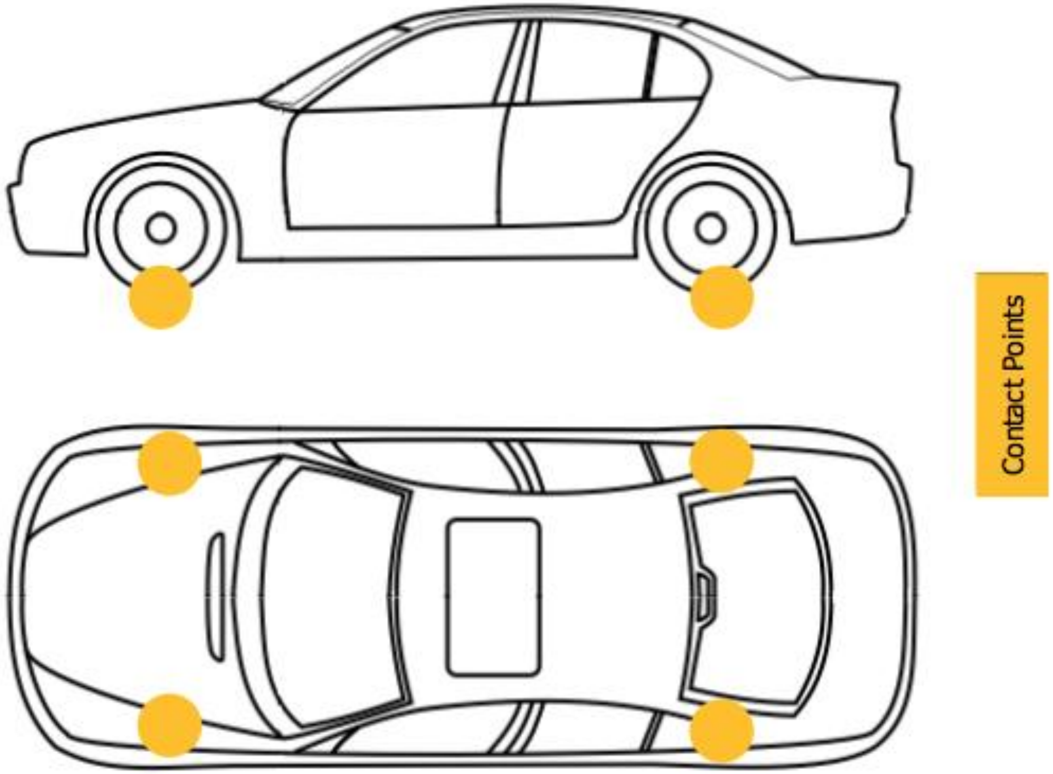
- The OpenDRIVE® file format provides the following features:
  - XML format
  - hierarchical structure
  - analytical definition of road geometry (plane elements, lateral / vertical profile, lane width etc.)
  - various types of lanes
  - junctions and junction groups
  - logical inter-connection of lanes
  - signs and signals incl. dependencies
  - signal controllers (e.g. for junctions)
  - road surface properties (OpenCRG)
  - road and road-side objects
  - user-definable data beads
  - etc.



OpenDRIVE

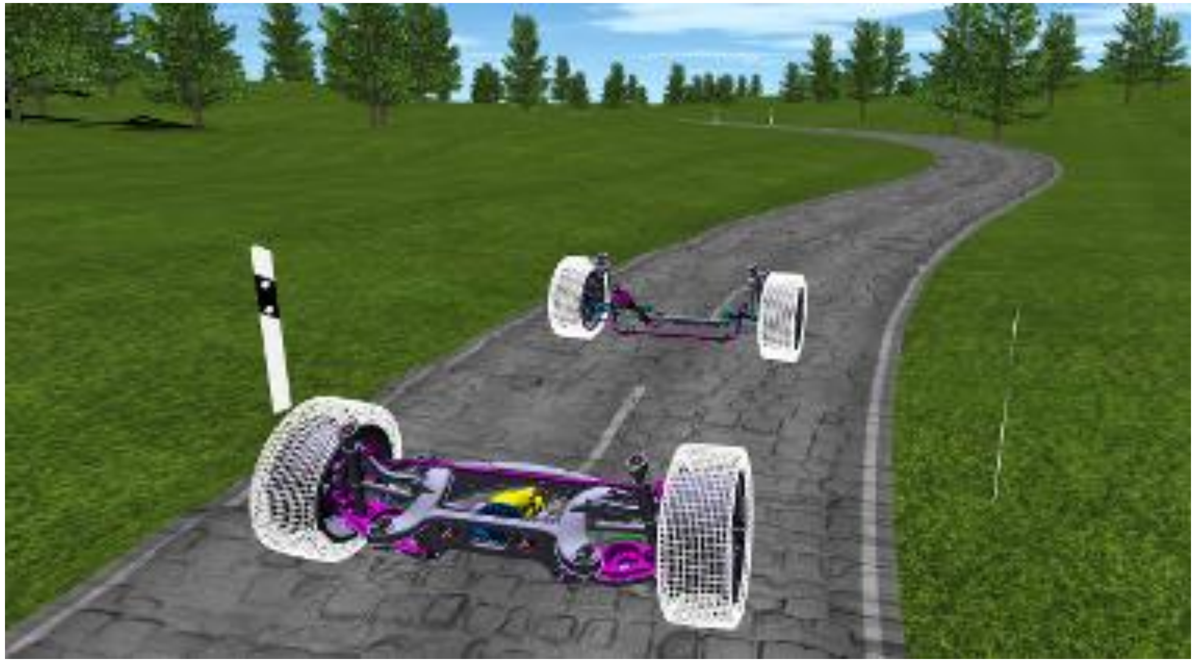
- The OpenDRIVE Manager (OdrManager) is a real-time compliant library enabling the user to read OpenDRIVE networks and to perform queries on them.
- In addition to the road logics contained in the OpenDRIVE format, the manager may also evaluate detailed information about a road surface which may be available in OpenCRG format.





OpenCRG

- The OpenCRG® initiative was launched in Oct. 2008.
- Its objective is the provision of a series of **open file formats and open source tools** for the detailed description, creation and evaluation of road surfaces.
- It is suitable for a broad range of applications including e.g. tyre and driving simulations.
- OpenCRG® is to be understood as a complementary project to OpenDRIVE®.
- Whereas the latter offers a macroscopic description of road networks, OpenCRG® will provide everything necessary for the microscopic view of road surfaces.

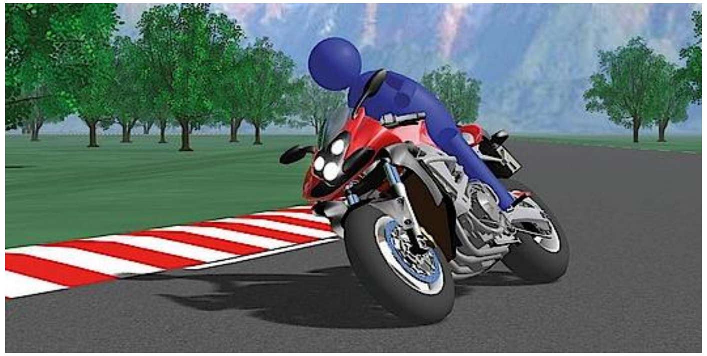


OpenCRG

- The OpenCRG® project provides the following features:
  - various ASCII/binary file formats with clear-text headers
  - handling of arbitrary scalar data vs. a reference grid (typically: elevation, friction coefficients etc.)
  - open source C-API for data handling and evaluation
  - open source MATLAB® API for data manipulation and generation
  - growing library of sample data
  - etc.
- OpenCRG® data sets are designed to describe patches of road surfaces in a very detailed manner, so that they may be used for the followin applications:
  - tire simulation
  - vibration simulation
  - driving simulation
  - etc.

IPG Automotive

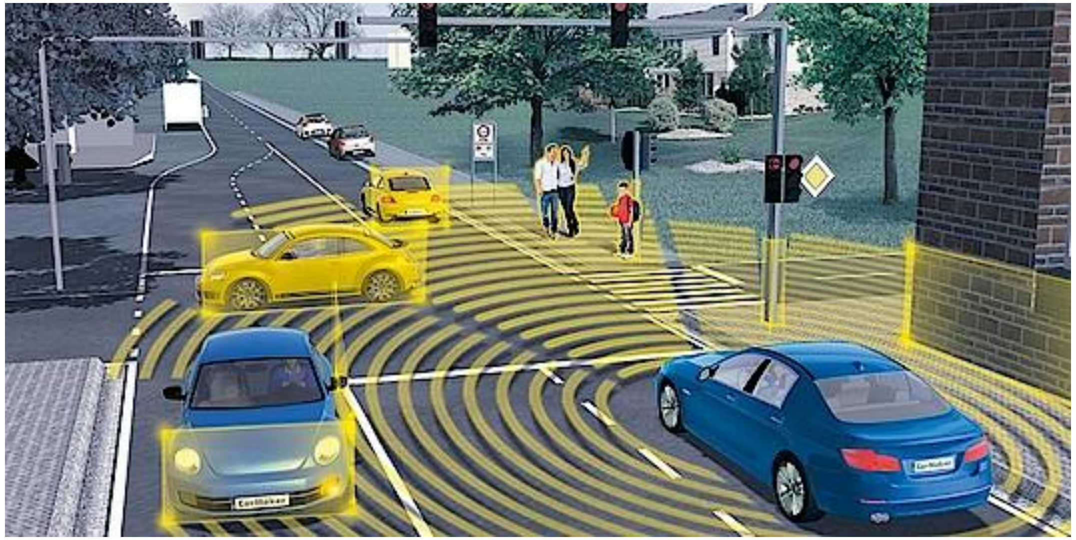
- The aim is to ensure **virtually** testing systems in the whole vehicle in realistic traffic situations, using an approach that is accurate in every detail.
- By using **virtual** test driving in addition to real-world test driving, it can increase the efficiency of the development process considerably, thereby achieving significant reductions in costs and time.
- Use front-loading to test innovations at an early stage in the system network of the whole vehicle and in every conceivable traffic scenario.
- **CarMaker**, **TruckMaker**, **MotorcycleMaker** simulation solutions as open integration and test platforms.
- The CarMaker product family features a high-precision, real-time capable model environment, which enables to create realistic and detailed emulations of real-world test runs and test scenarios tailored entirely to individual requirements.
- Visualization tools for the clear and reliable analysis of the test run and the possibility of test automation and automated analysis complete comprehensive package for virtual test driving.



IPG Automotive

- ADAS need to function in every conceivable traffic scenario. Do the right maneuver at the right time.
- As the number of assistance functions continues to grow, this leads to increased interaction with other vehicle systems.
- Their automotive systems engineering approach is a clearly formulated motivation for helping customers find suitable solutions for the development of systems.
- Virtual prototypes allows to conduct early testing on individual functions or system networks without a real prototype, yet in realistic traffic scenarios within the context of the whole vehicle.
- Front-loading can also help to achieve greater system maturity, since ADAS need to be able to initiate driving functions independently and intervene in driving maneuvers.
- Reproduce the relevant factors for tests in the virtual world: from sensor models through road networks to other road users and much more.
- Use virtual test driving as a complement to real-world tests to allow to bring systems into series production within very short time frames.

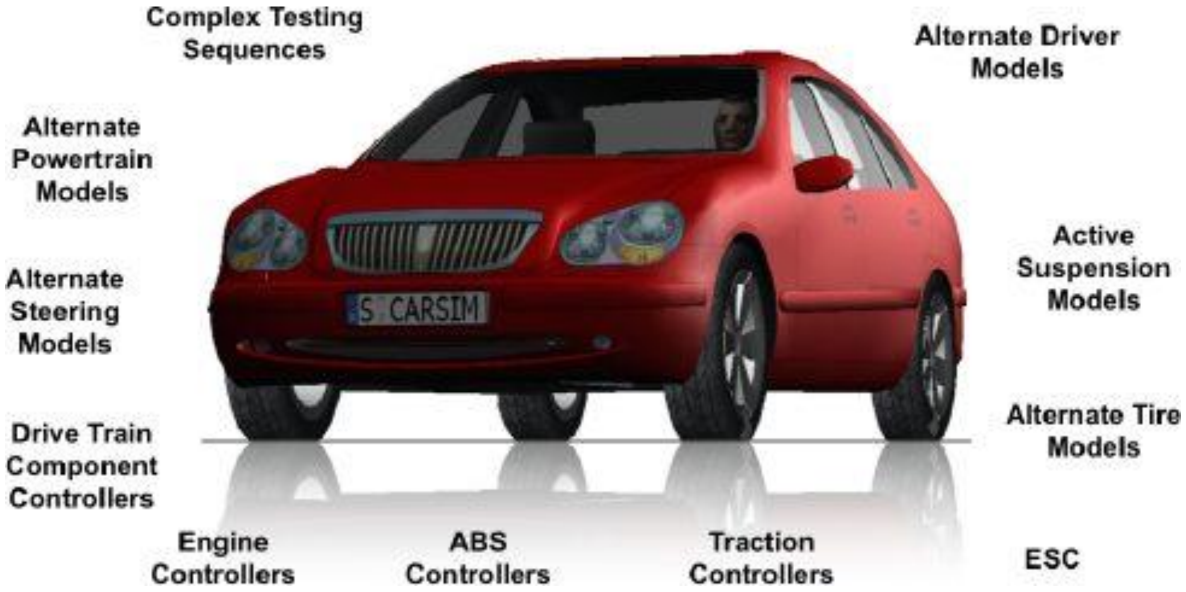
IPG Automotive



Mechanical Simulation

Mechanical Simulation®
Driving Simulation.

- It produces and distributes SW tools for simulating and analyzing the dynamic behavior of motor vehicles in response to steering, braking, and acceleration control inputs.
- These software tools are extensively validated and correlated to real-world results as measured and observed by many automotive OEMs.
- The vehicle simulation packages allow to extend the models using the built-in VS Command language and/or external software such as MATLAB/Simulink, LabVIEW, C code, and others.
- They can be run in real-time with **Hardware in the Loop** (HIL) or as driving simulators.



carSIM

bikeSIM

truckSIM

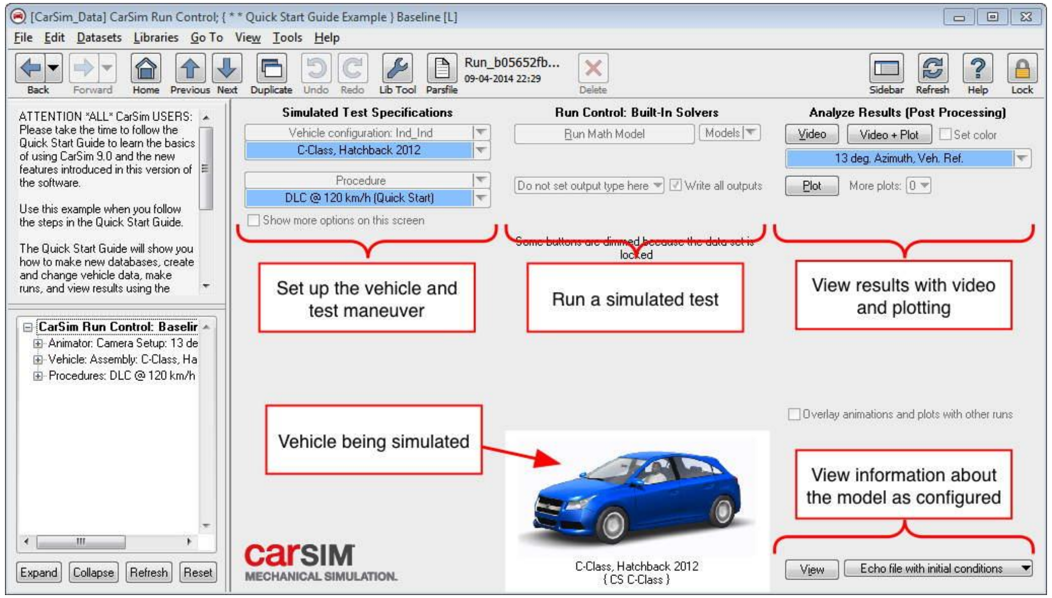
suspensionSIM

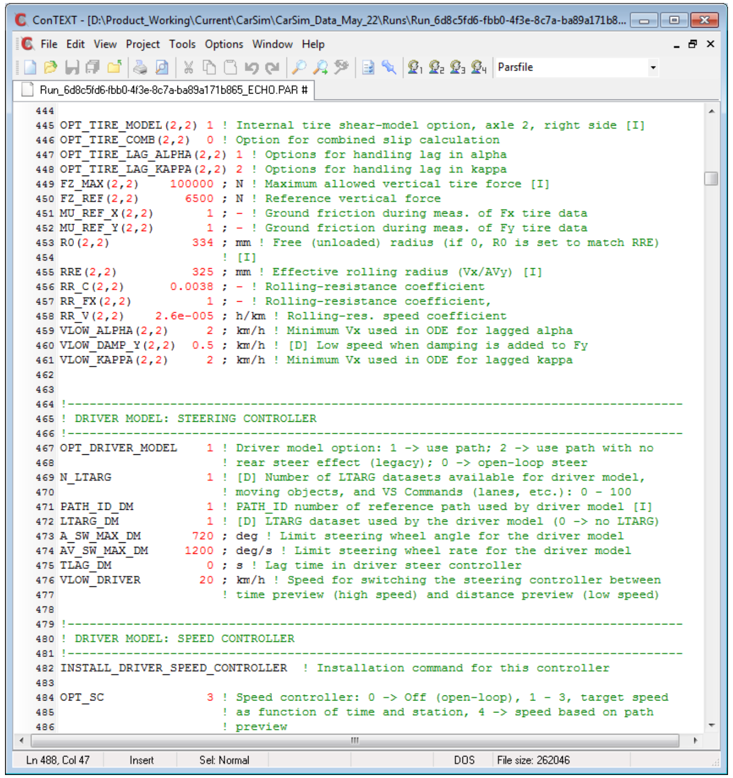
Mechanical Simulation

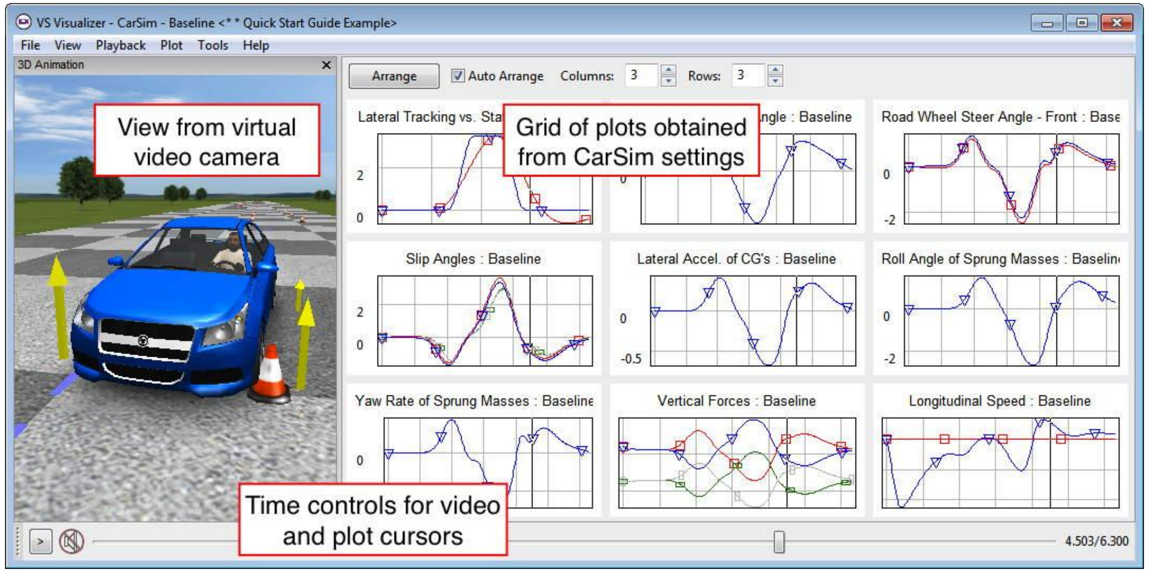
Mechanical Simulation®

Driving Simulation.

- VehicleSim products include 3 major types of programs:
  - VS **Browser**: graphical user interface (GUI).
  - VS **Solvers**: response of math models in simulation.
  - VS **Visualizer**: simulation using animation and plotting.

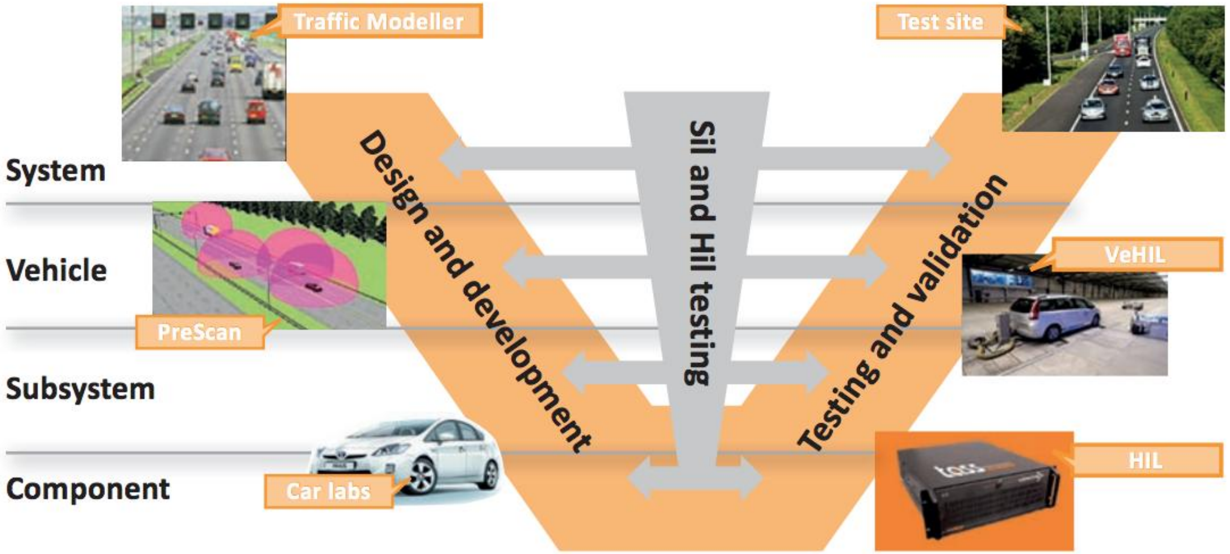






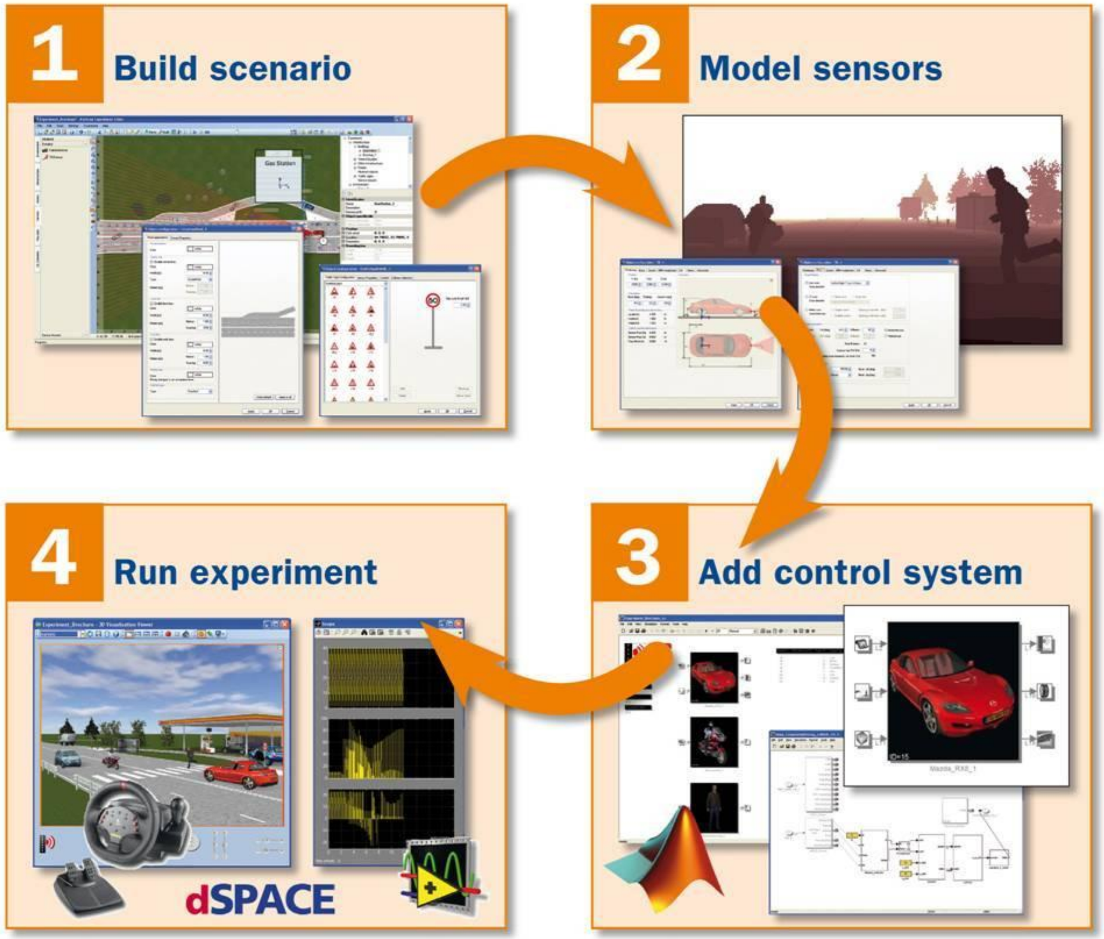
Tass: A Dutch company in Siemens

- Purchased by **Siemens** in Aug. 2017.
- Simulation and validation in automated driving:
    - Development phase: a **simulation platform called PreScan** for developing active safety systems, automated driving systems and connected vehicle systems.
    - Testing: **hardware in the loop testing** is necessary for sensor and communication systems, while a **dedicated test environment** for rapid, safe and reproducible testing of cooperative and automated vehicles.
    - Validation and performance testing: a **test site for urban, interurban and highway** traffic is available.



Tass: A Dutch company in Siemens

- PreScan is a simulation platform consisting of a GUI-based preprocessor to define scenarios and a run-time environment to execute them.
- PreScan can be used from model-based controller design (MIL) to real-time tests with software-in-the-loop (SIL) and hardware-in-the-loop (HIL) systems.
- It is an open software platform with flexible interfaces to link to 3rd party vehicle dynamics model (e.g. CarSIM, dSPACE ASM) and 3rd party HIL simulators/hardware (e.g. ETAS, dSPACE, Vector).
- PreScan comprises several modules that provide everything an ADAS system developer needs.
- An intuitive GUI allows to build the scenario and model sensors, while the Matlab/Simulink interface enables to add a control system.
- In the experiment, the visualisation viewer gives a realistic 3D representation of the scenario.

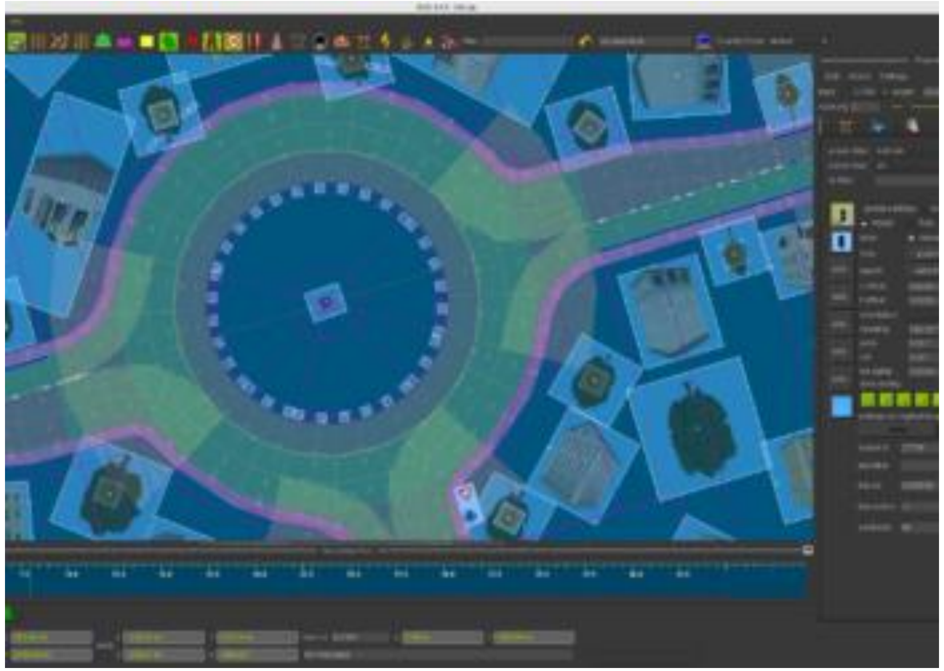


VIRES: A German company in MSC

- VIRES Simulations technologie GmbH, provides simulation solutions for the automotive, railroad and aerospace industries.
- VIRES is a member of the **MSC Software** Group of companies and is owned by Hexagon.
- "**VIRES Virtual Test Drive**"(**VTD**) is used for the development and testing of advanced driver assistance and active safety systems, leading to solutions for automated driving.
- VTD is a complete tool-chain for driving simulation applications.
- VTD is a toolkit for the creation, configuration, presentation and evaluation of virtual environments in the scope of road and rail based simulations.
- It is used for development of ADAS and automated driving systems and the core for training simulators.
- It covers the full range from the generation of 3d content to the simulation of complex traffic scenarios and, finally, to the simulation of either simplified or physically driven sensors.
- It is used in SiL, DiL, ViL and HiL applications and may also be operated as co-simulations including 3rd party or custom packages.

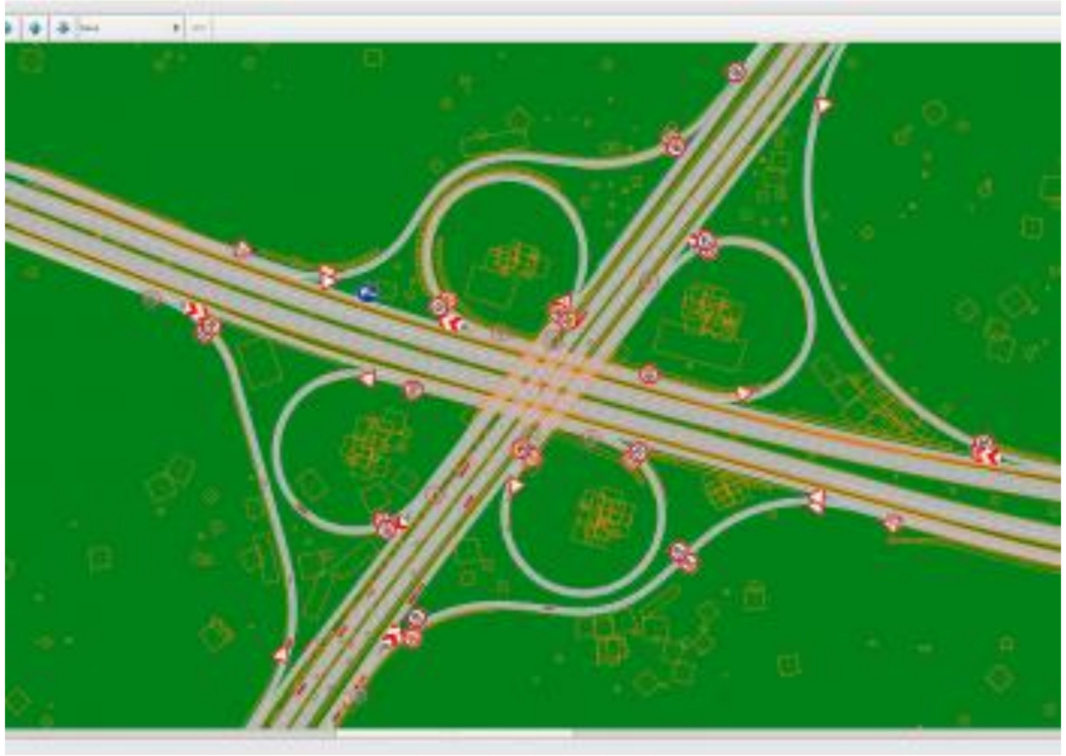
VIRES in MSC: VTD

- The interactive Road Network Editor (ROD) allows to design road and rail networks in full detail with unlimited numbers of lanes, complex intersections, comprehensive signs and signaling. It links and exports logic and graphic data consistently from a single source.



VIRES in MSC: VTD

- Dynamic content is defined with an interactive scenario editor.
- It visualizes the underlying **OpenDRIVE®** database and allows the user to specify traffic as individual objects and as autonomous swarms around key entities.
- The large library of vehicles, pedestrians and driver properties may easily be customized.



VIRES in MSC: VTD

- VTD will seamlessly adapt, no matter what environment in (XiL), what time-base (real-time or non real-time) or which additional components (co-simulations);
- VTD may be operated from a single computer up to a full-scale HPC environment.

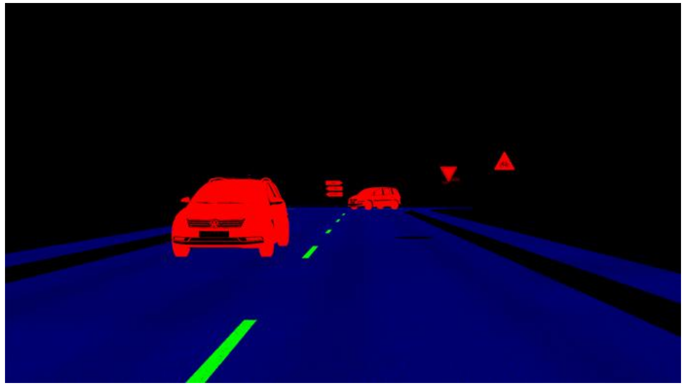
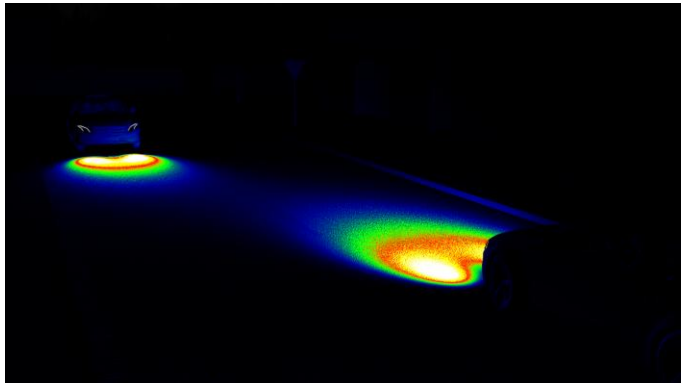
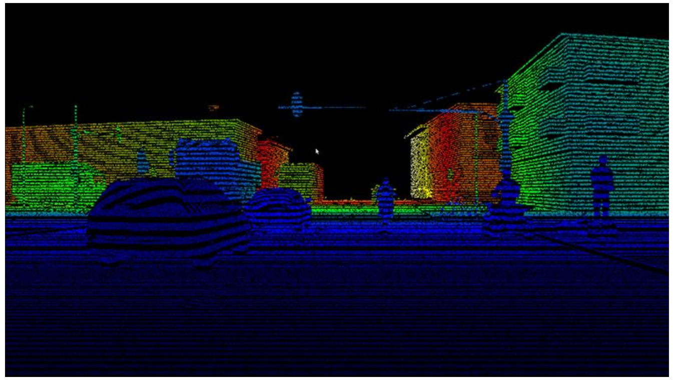




VIRES in MSC: VTD

- VTD is an out-of-the-box solution but doesn’t stick to this box.
- The user may customize VTD with SDKs along with ready-to-go templates for sensor simulation (object-list based and physics based), dynamics simulation and image generation.
- The open interfaces for run-time data and simulation control make it easy to integrate VTD.



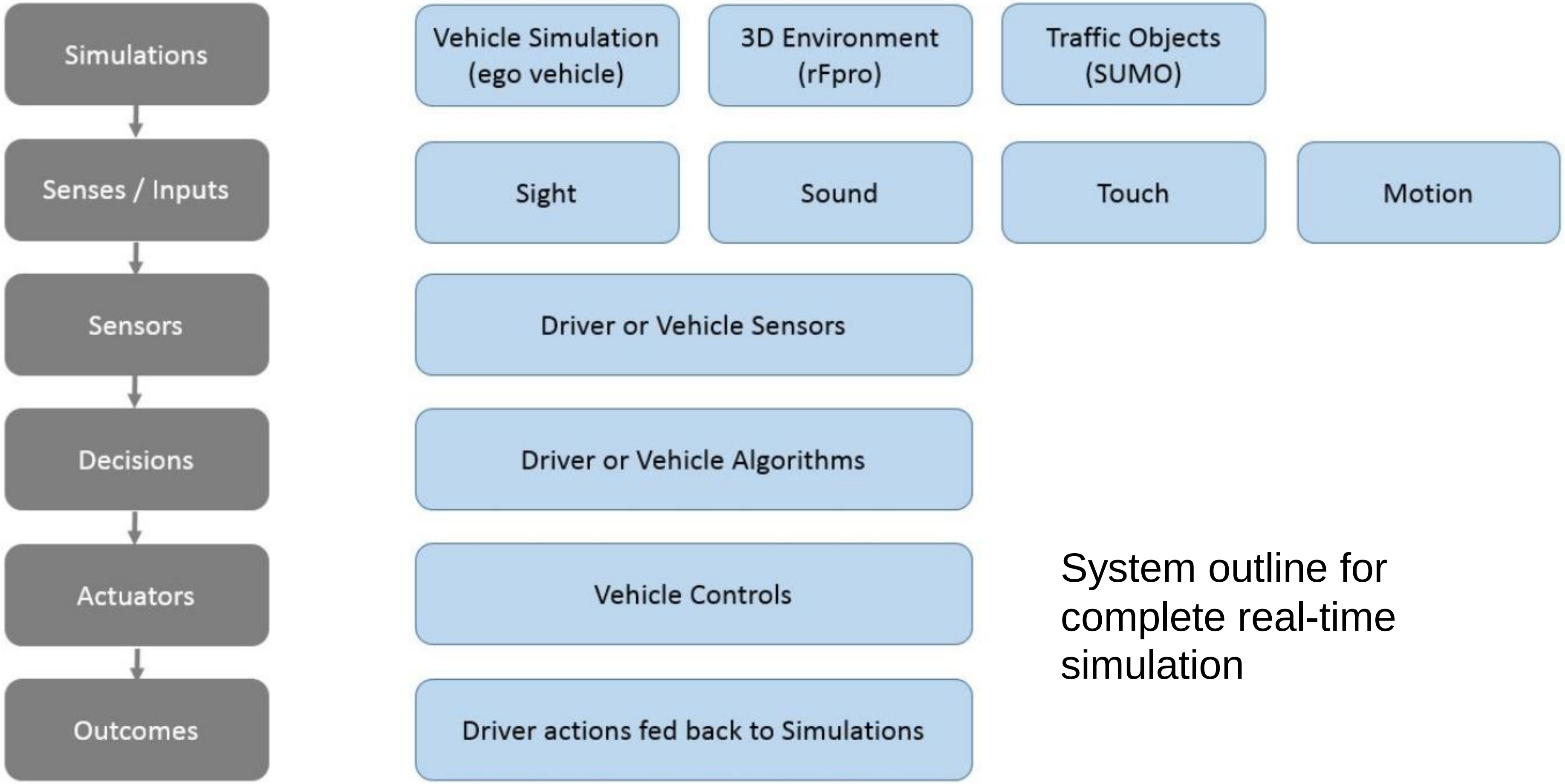
rFpro: An UK company

- Driving Simulation
  - Used for driving simulation by 6 of the top 10 OEMs and many smaller OEMs and T1s for Virtual Test programmes of vehicles, subsystems, ADAS and Autonomous control systems;
- It focuses on driving simulation for the engineering development of vehicle dynamics, ADAS and Autonomous control systems.;
- ADAS: driving simulator or Engineer's Workstation can be used to test emergency scenarios with a human test driver, in complete safety.
  - Running sensor models, algorithms and control systems in Engineer's Workstation or driving simulator, benefit from Digital Road Models, built to extreme levels of accuracy, allowing to conduct tests in simulation, correlating very closely to real-world testing along the same public road test routes.
  - By controlling virtual conditions such as weather, time of day, time of year, turbidity, quality of road markings, large amounts of testing can be done on extremely accurate copies of the real world, validating and calibrating algorithms.

rFpro: An UK company

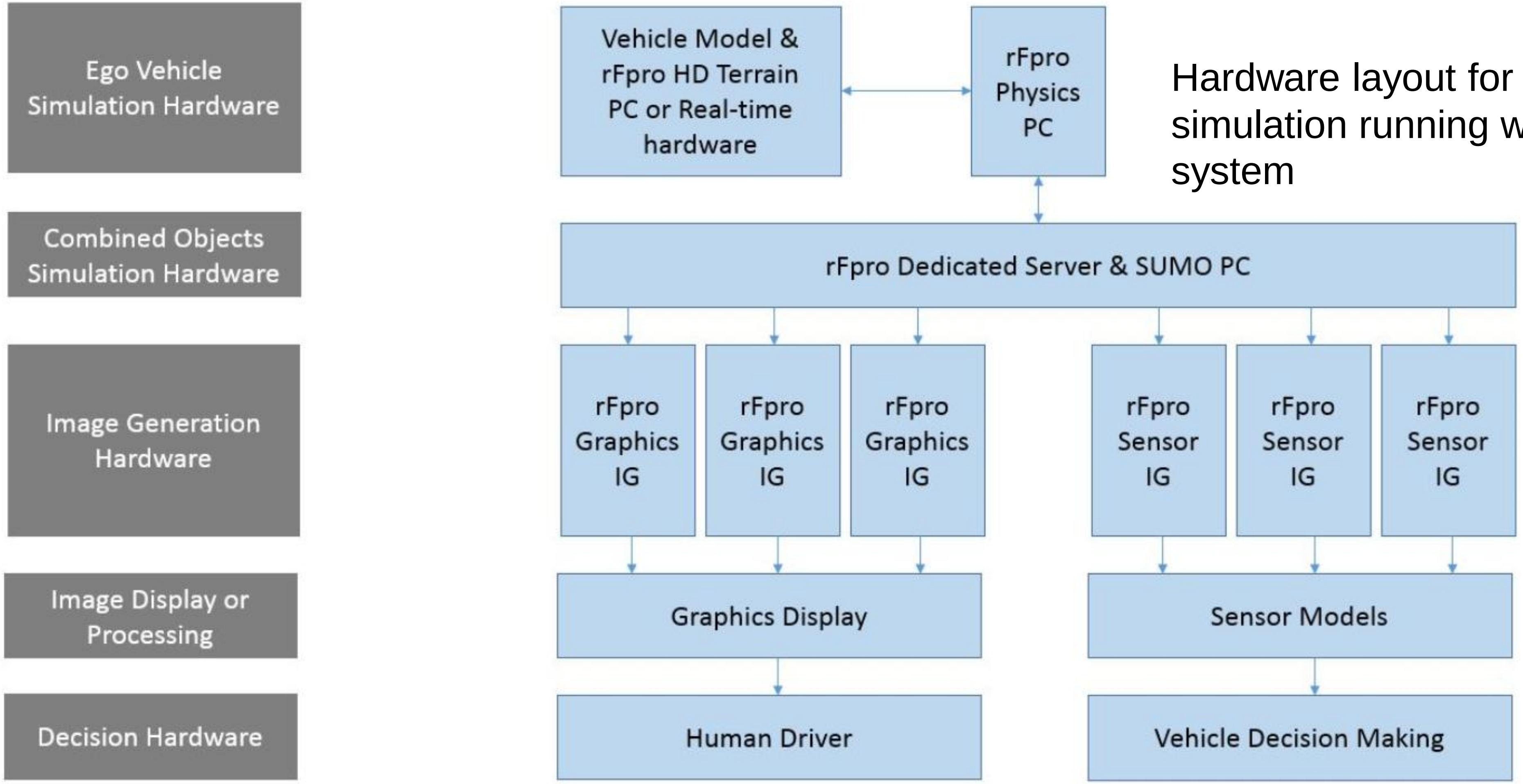
- Traffic and pedestrians:
  - Weather and Traffic interfaces are deterministic, so repeatable tests and experiments can be created, or repeated, to train deep learning systems and to provide a regression testing environment for AI driver models.
- High quality LiDAR scans, data processing and modellers.
- Road model: build race and test tracks covering the major race series, F1, NASCAR, IndyCar, WEC, DTM, IMSA, Formula E.
  - Build real road testing routes & proving grounds across the continents.
  - Automated road building pipeline based on the use of kinetic LiDAR (laser scanning) technology since 2008.
  - LiDAR survey with geo-referenced spherical photography.
  - Run in the Cloud on rented AWS servers to create the road surfaces and reference material used for the scenery in the completed 3D digital models.

rFpro: An UK company



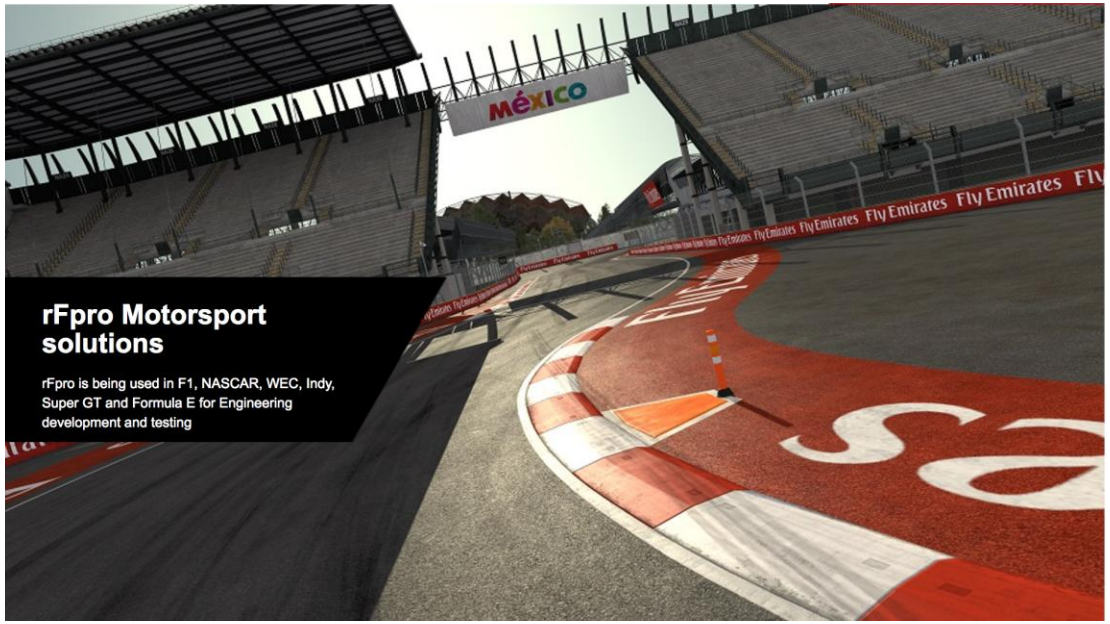
System outline for complete real-time simulation

rFpro: An UK company



Hardware layout for a real-time simulation running with a multi-channel system

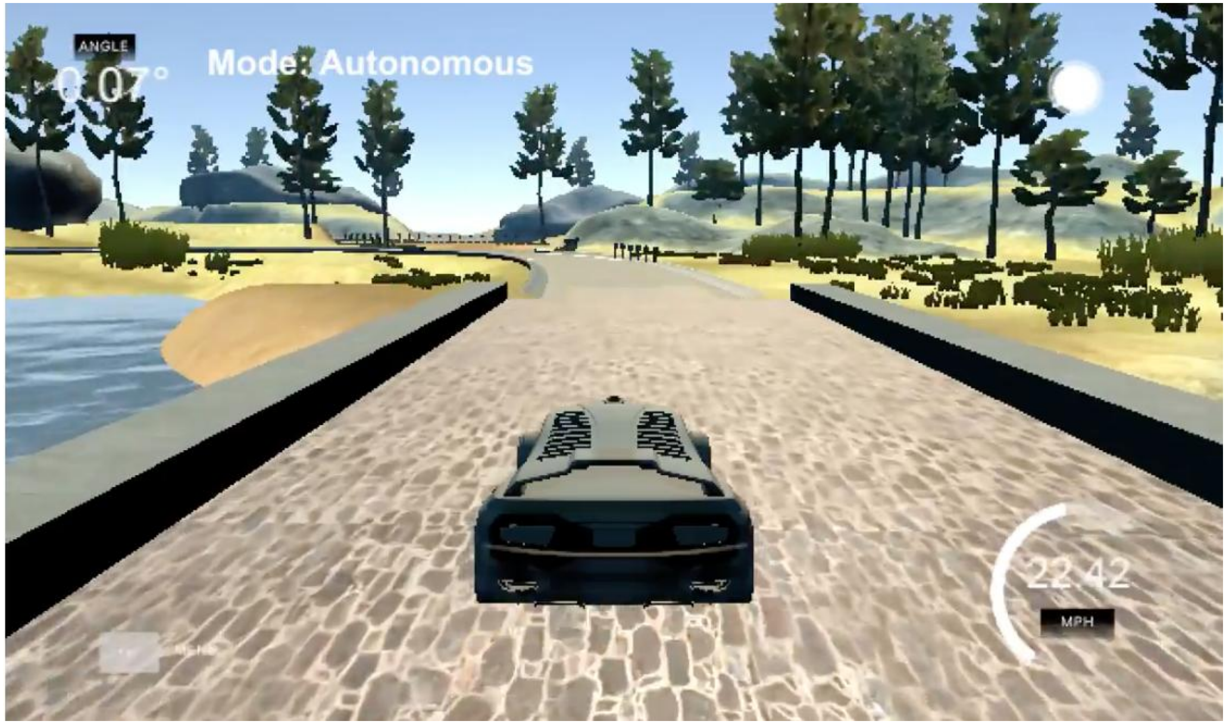
rFpro: An UK company



A self-driving car simulator with Unity at Udacity

- Open source: https://github.com/udacity/self-driving-car-sim;
- Training Mode or Autonomous Mode;
- In the training mode, driving the car manually to record the driving behavior, then use the recorded images to train a machine learning model.
- In the autonomous mode, testing the machine learning model to see how well the model can drive the car without dropping off the road;
- Technically, the simulator is acting as a server where the program can connect to and receive a stream of image frames from.
- A program can use a machine learning model to process the road images to predict the best driving instructions, and send them back to the server.
- Each driving instruction contains a steering angle and an acceleration throttle, which changes the car’s direction and the speed (via acceleration). As this happens, the program will receive new image frames at real time.

A self-driving car simulator with Unity at Udacity

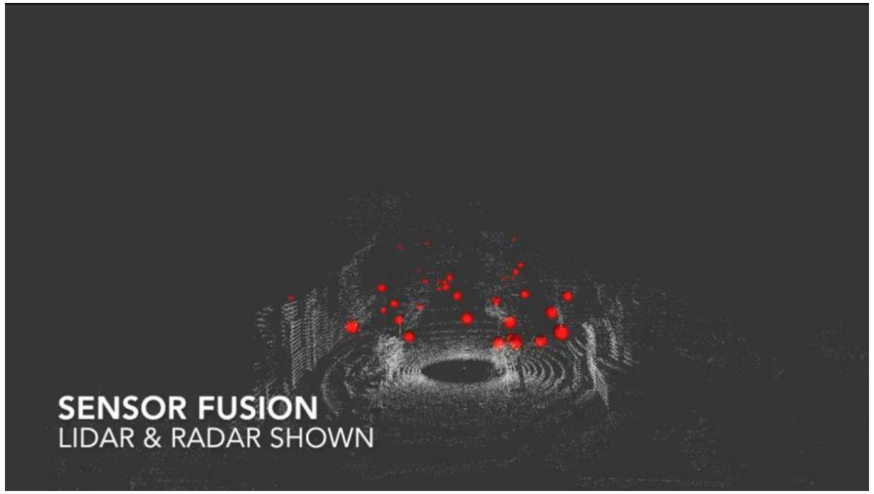


It is rewarding to watch a trained model autonomously driving the car without going off the road.

RightWorld at RightHook

- A Palo Alto startup: www.righthook.io;
- RightWorld: features a flexible tool chain for environment creation from HD maps, a robust traffic simulation, rich scenario creation, and large-scale grid/cloud management;
- RightWorldHIL: a complete end-to-end testing solution for ADAS HW;
- RightWorldHD: Software testing of virtual drivers by simulating vehicle dynamics, environment and sensor models in real time or faster.
- RightReplay allows replay and scenario generation from existing datasets collected in the real world, turning a once-in-a-lifetime event into a permanent test case.

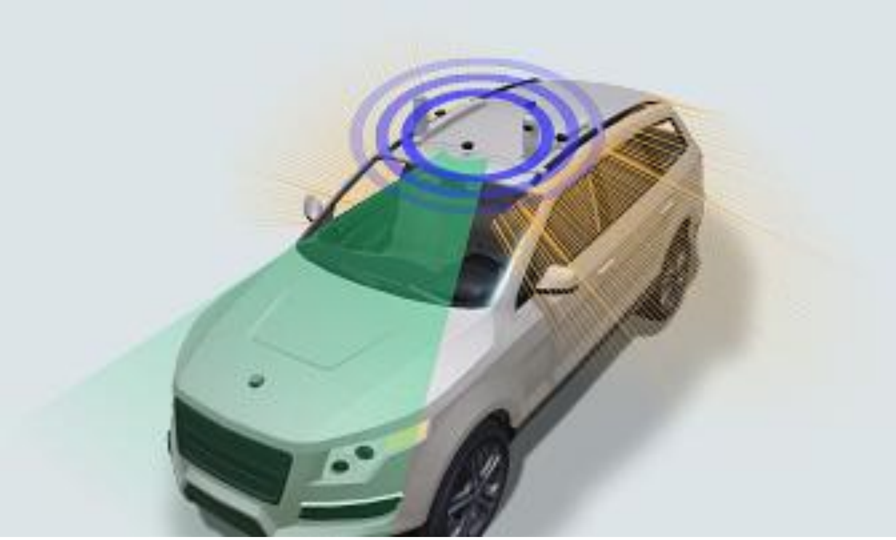
RightWorld at RightHook



Cogna's Deep Learning Simulation Engine

- A Isarel company: http://www.cognata.com/;
- A deep learning simulation engine leverages city mesh combined with DNN and AI.
- Its VR simulator and engine enable autonomous car manufacturers to run thousands of different scenarios based on geographic locations and driver behaviors.
- TrueLife: Adding dynamic layer of traffic model of other vehicles and pedestrians.
- PhysicsStudio: Local weather conditions and lighting are added to stress test the system.
- Combining TrueLife and PhysicsStudio to simulate the sensor interaction with the external materials including sensors ,saturation and lost packets to receive autonomous driving simulation feedback loop.
- Each mile driven on the simulator is equal to miles driven regularly as creating use cases that train the AI driving system and reduce the time to reach maturity.
- It uses computer vision and deep learning to automatically generate a whole city-simulator including buildings, roads , lane marks, traffic signs and trees and bushes.

Cogna's Deep Learning Simulation Engine



Cvedia-SynCity

- Generate synthetic training data with sensor simulations incorporating real-world phenomena such as day/night cycles, weather and lighting conditions.
- Photorealistic rendering of our world enabling high level of transfer learning for autonomous vehicles and robotic applications.



Cvedia-SynCity

- Sensor simulations are engineered at protocol level: LiDAR, radar, near and far IR, 360-degree cameras, IMU, AHRS and GNSS/GPS.
- Compatible with ROS/ROS2, SynCity is a combined SIL & HIL solution.
- A scalable hardware deployment that connects directly with your external HW.

Thanks!