AutonoVi-Sim: Autonomous Vehicle Simulation Platform with Weather, Sensing, and Traffic Control

Andrew Best[1] Sahil Narang[1] Lucas Pasqualin[2] Daniel Barber[2] Dinesh Manocha[1]
[1]Department of Computer Science, UNC Chapel Hill [2]Institute for Simulation and Training, UCF
{best,sahil,dm}@cs.unc.edu {lpasqual,dbarber}@ist.ucf.edu

Abstract

We present AutonoVi-Sim, a novel high-fidelity simulation platform for autonomous driving data generation and driving strategy testing. AutonoVi-Sim is a collection of high-level extensible modules which allows the rapid development and testing of vehicle configurations and facilitates construction of complex traffic scenarios. Autonovi-Sim supports multiple vehicles with unique steering or acceleration limits, as well as unique tire parameters and dynamics profiles. Engineers can specify the specific vehicle sensor systems and vary time of day and weather conditions to generate robust data and gain insight into how conditions affect the performance of a particular algorithm. In addition, AutonoVi-Sim 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. Autonovi-Sim facilitates training of deep-learning algorithms by enabling data export from the vehicle's sensors, including camera data, LIDAR, relative positions of traffic participants, and detection and classification results. Thus, AutonoVi-Sim allows for the rapid prototyping, development and testing of autonomous driving algorithms under varying vehicle, road, traffic, and weather conditions. In this paper, we detail the simulator and provide specific performance and data benchmarks.

1. Introduction

Autonomous driving represents an imminent challenge encompassing a number of domains including robotics, computer vision, motion planning, civil engineering, and simulation. Central to this challenge are the safety considerations of autonomous vehicles navigating the roads surrounded by unpredictable actors. Humans, whether drivers, pedestrians, or cyclists, often behave erratically, inconsistently, or dangerously, forcing other vehicles (including autonomous vehicles) to react quickly to avoid hazards. In order to facilitate acceptance and guarantee safety, vehicles must be tested not only in typical, relatively safe scenarios, but also in dangerous, less frequent scenarios.

Aside from safety concerns, costs pose an additional challenge to the testing of autonomous driving algorithms. Each new configuration of a vehicle or new sensor requires re-calibration of a physical vehicle, which is labor intensive. Furthermore, the vehicle can only be tested under conditions limited either by a testing track, or the current traffic conditions if a road test is being performed. This means the vehicle can be tested no faster than real-time and without any speedups or parallel testing.

Many recent approaches to autonomous driving rely on machine-learning via Bayesian networks or deep-learning to provide entity detection [23], entity prediction [13], and end-to-end control [6]. However, such approaches rely on substantial amounts of annotated data in safe, as well as dangerous scenarios. The dataset must also encompass varied weather and lighting conditions. In addition, not all autonomous vehicles are equipped with identical or equivalent sensing capability; training data must be available for the specific configuration or sensors of the vehicle being tested. Gathering such data by physical tests can be expensive, difficult and even dangerous. In contrast, a high-fidelity simulator can augment and improve training of algorithms, and allow for testing safely and efficiently.

Insights gained from simulation could provide critical training data and information on algorithmic inefficiencies before actual vehicle testing. In an effort to facilitate progress in these areas, we present AutonoVi-Sim, a simulation framework for training and testing autonomous driving algorithms and sensors. AutonoVi-Sim is a collection of high-level, extensible modules designed to allow researchers and engineers to rapidly configure novel road networks, driving scenarios, and vehicle configurations, and to test these in a variety of weather and lighting conditions. AutonoVi-Sim captures a variety of autonomous driving phenomena and testing requirements including:

- **Data Generation**: Autonovi-Sim facilitates data anal-

ysis by allowing exports of relevant data for traffic proximate to the autonomous vehicle as well as data from each virtual sensor on the vehicle. Sensor and local traffic data can be used in training deep-learning approaches by generating automatically labelled classification and decision data efficiently.

- **Varying vehicle, cyclist, pedestrian, and traffic conditions**: AutonoVi-Sim includes various vehicle and sensor models, pedestrians, and cyclists. Diversity of these traffic entities allows for training classification on differing shapes, sizes, colors, and behaviors of cyclists, pedestrians, and other drivers.

- **Dynamic Traffic, Weather and Lighting Conditions**: AutonoVi-Sim provides high fidelity traffic simulation, supporting dynamic changes in traffic density, time of day, lighting, and weather including rain and fog.

- **Rapid Scenario Construction**: Typical road networks can be easily laid out using spline painting and are automatically connected for routing and navigation purposes. AutonoVi-Sim supports many lane configurations and atypical road geometry such as cloverleaf overpasses. In addition, other vehicles and entities can be scripted to generate repeatable erratic behavior, e.g. cutting in front of the ego-vehicle, walking into the road.

The rest of the paper is organized as follows. In section 2, we motivate simulation as a tool for advancing autonomous driving and detail related work in the field. In section 3, we detail the core modules provided by AutonoVi-Sim. We reserve discussion of the *Drivers* and *Vehicles* modules for section 4 and offer demonstrations of the simulator in section 5.

2. RELATED WORK

Simulation has been an integral tool in the development of controllers for autonomous vehicles. [12], [18], and [27] offer in-depth surveys of the current state of the art and the role simulation has played. Many successful vehicle demonstrations of autonomy were first tested in simulation [2, 21, 1]. Recent work in traffic modelling has sought to increase the fidelity of the modelled drivers and vehicles; a survey is provided in [8].

Recent studies support the use of high-fidelity microscopic simulation for data-gathering and training of vision systems. [25] and [17] leveraged Grand Theft Auto 5 to train a deep-learning classifier at comparable performance to manually annotated real-world images. Several recent projects seek to enable video games to train end-to-end driving systems, including ChosenTruck and DeepDrive-Universe which leverages the OpenAi Universe system. Using video game data provides benefits in the fidelity of the vehicle models but limits the ability to implement sensing systems and access data beyond visual data. A fully dedicated high-fidelity simulator can address these limitations and provide access to point-cloud data, visual data, and other vehicle sensors without the limitations imposed by adapting gaming software. Research in this area has begun to emerge [10]. Our work is complimentary to such systems and can be combined with generated data from other simulators to increase robustness of training data.

Modeling Vehicle Kinematics and Dynamics: A number of approaches have been developed to model the motion of a moving vehicle, offering trade-offs between simplicity, efficiency and physical accuracy of the approach. Simpler models are typically based on linear dynamics and analytical solutions to the equations of motion [20]. More accurate models provide a better representation of the physical motion, but require more computational power to evaluate and incorporate non-linear forces in the vehicle dynamics [7]. Margolis and Asgari [22] present several representations of a car including the widely used single-track bicycle model. Borrelli et al. [7] extend this model by including detailed tire-forces. The Reeds-Shepp formulation is a widely used car model with forward and backward gears [24]. Our simulator leverages the NVIDIA PhysX engine for the underlying vehicle model.

Modeling Traffic Rules: As well as planning the appropriate paths to avoid collisions, autonomous vehicles must also follow applicable laws and traffic norms. Techniques have been proposed to simulate typical traffic behaviors in traffic simulation such as Human Driver Model [29] and data-driven models such as [16]. Logic-based approaches with safety guarantees have also been demonstrated [30]. An extensive discussion on techniques to model these behaviors in traffic simulation can be found in [8]. Our simulator allows for modelling such trafic behaviors as well as traffic control strageties at the infrastructure level.

Path Planning and Collision Avoidance: Prior approaches to path planning for autonomous vehicles are based on random-exploration [11], occupancy grids [19], potential-field methods [14], driving corridors [15], etc. Recent approaches seek to incorporate driver behavior prediction in path planning using Bayesian behavior modeling [13] and game-theoretic approaches [26]. Continuous approaches for collision-avoidance have been proposed based on spatial decomposition or velocity-space reasoning. Ziegler et al. [32] utilize polygonal decomposition of obstacles to generate blockages in continuous driving corridors. Sun et al. [28] demonstrate the use of prediction functions and trajectory set generation to plan safe lane-changes. Berg et al. [31] apply velocity-space reasoning with accel-

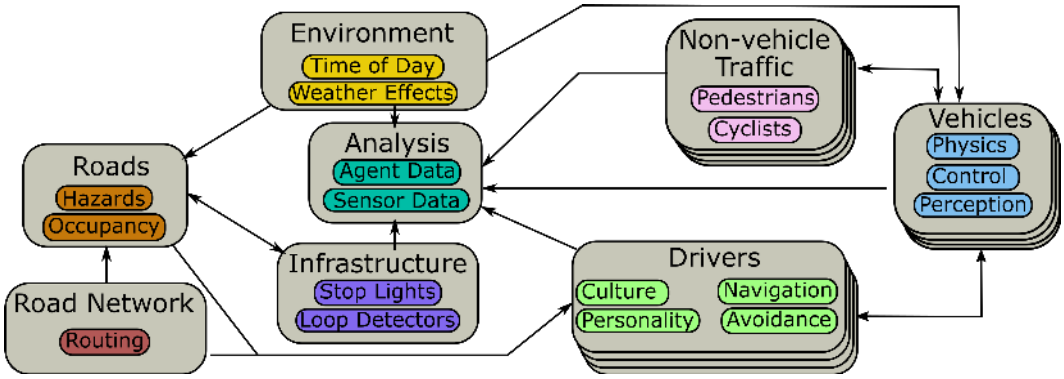


Figure 1. **AutonoVi-Sim Platform Overview:** The eight modules composing AutonoVi-Sim encompass varying aspects of autonomous driving. The *Road*, *Road Network*, and *Infrastructure* modules define the driving environment. The *Environment* module allows engineers to specify specific environment conditions 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. Finally, the *Analysis* module is used to catalog and export data, including agent positions and sensor readings, for analysis.

eration constraints to generate safe and collision-free velocities. Bareiss et al. [4] extend the concept of velocity obstacles into the control space to generate a complete set of collision-free control inputs. We have implemented several driving strategies in the simulator to demonstrate its generality.

3. SIMULATION MODULES

Drawing from recent work in crowd simulation, [9], AutonoVi-Sim is divided into eight extensible modules, each with various sub-components. The modules are Environment, Road Network, Road, Drivers, Infrastructure, Vehicles, Non-vehicle Traffic, and Analysis. Each module captures some aspect of autonomous driving simulation and can be extended and modified to suit the specific needs of a particular algorithm. Figure 1 shows the connection between components in AutonoVi-Sim. In this section, we will detail the modules which make up the basic simulation system, reserving discussion of the vehicle and driving strategy modules for section 4.

3.1. Roads

Roads in AutonoVi-Sim are represented by their center line, a number of lanes and directions thereof, and the surface friction of the road. Roads are placed interactively by drawing splines on a landscape which allows quick construction. Each road maintains occupancy information, average flow, and can maintain hazard information. The road module also maintains the set of hazards such as potholes or debris, which can be specified by density (number of hazards per km) or interactively by placing them on the road.

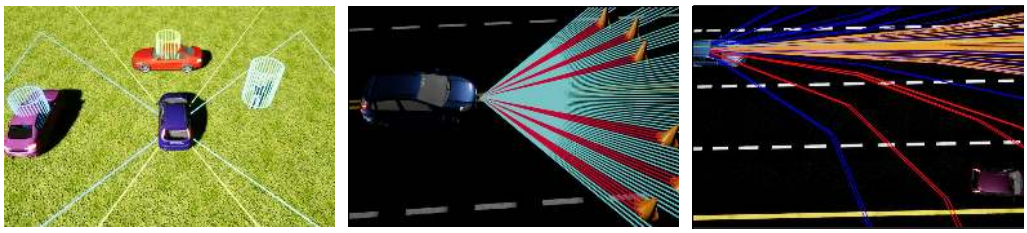
Alternately, roads can be specific pieces of geometry as in the case of intersections. This provides the flexibility to place specific intersections and model atypical road constructions for modelling specific environments. Figure 2(A) shows an example of road placement in AutonoVi-Sim.

3.2. Infrastructure

Infrastructure controllers represent traffic lights, signage, and any other entity which modifies the behaviors of vehicles on the road. These controllers can be added specifically to roads, as in the case of intersections, or placed independently as in signage or loop detectors. Vehicles implement their own detection of these entities as is described in section 4.1.2. Infrastructure components are provided with basic detection capability and agency. For example, traffic lights can determine which lanes are congested and adjust the light cycle accordingly to more traffic more effectively.

3.3. Road Network

The road network in AutonoVi-Sim provides the basic connectivity information for the traffic infrastructure to the vehicles in the simulation. At run-time, the network is automatically constructed by connecting roads into a directed graph. The road network provides GPS style routing to vehicles and localization for mapping purposes. Coupled with the road and infrastructure modules, the Road Network also provides information about upcoming traffic and current road conditions. As part of the Road Network, vehicle spawners are provided which generate vehicles and can provide specific destinations for each vehicle. The Road Network can be used to specify per-road initial density as well or to specify a general initial traffic density over the network.



(A) (B) (C)

Figure 2. **AutonoVi-Sim Sensor Modules: (A):** Sensors on the vehicle are placed interactively. The most basic sensors (shown here) provide ray-cast detection with configurable noise, detecting time, angle, and range. The sensors here are configure to provide debugging detection information. **(B)**: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. **(C)**: Once sensors are placed, the vehicle's navigation algorithm can be tested and examined interactively. The driving algorithm described in [5] samples potential controls and projects forward in time. Red control paths indicate predicted collisions with the nearby vehicle. The data analysis module allows for exporting sensor data as the vehicle navigates.

3.4. Environment

The environment module allows engineers to specify the specific environmental conditions for a given driving scenario. This currently includes time of day and weather. The system implements varying levels of fog and rain conditions. Basic environmental effects such as road friction reduction are controlled by the environment module. The interaction between weather and specific sensors is implemented by the sensor module.

3.5. Non-Vehicle Traffic

AutonoVi-Sim implements two non-vehicle traffic participants: pedestrians and cyclists. Pedestrians operate separately from the road network and can be given specific destinations. By default, pedestrians follow safe traffic rules to navigate to their goal. They can also be setup to trigger specific occurrences. For example, as the ego-vehicle nears, a pedestrian can be triggered to walk into the street in front of the vehicle to test its reaction time.

Cyclists operate similarly to vehicles in AutonoVi-Sim. Cyclists are given destinations and route over the road network. Similarly to pedestrians, cyclists can be programmed to trigger erratic behavior under specified conditions. For example, as the ego-vehicle approaches, a cyclist can be triggered to stop in the road, suddenly change direction, or enter the road in an unsafe fashion.

3.6. Analysis and Data Capture

AutonoVi-Sim implements a module for logging positions, velocities, and behaviors of the various traffic participants. It also supports logging egocentric data from the vehicle, such as relative positions of nearby entities at varying times during simulation. Camera-based sensors can record out the video data captured during simulation as can LIDAR based sensors Section 4.1.2 describes sensors in more detail.

4. AUTONOMOUS DRIVING MODULES

The simulation modules described in section 3 serve as the basis for AutonoVi-Sim. This section describes the two core modules which allow for testing autonomous driving and sensing algorithms under varying conditions, the Drivers and Vehicles modules.

4.1. Vehicles

The vehicle in AutonoVi-Sim is represented as a physics-driven entity with specific tire, steering, and sensor parameters. Physics parameters include the base tire coefficient of friction, the mass of the vehicle, engine properties such as gear ratios, and the physical model for the vehicle. Each of these parameters can vary between vehicles and relevant properties such as tire friction or mass can vary at runtime as needed.

4.1.1 Control and Dynamics

Vehicle control is provided on three axes: steering, throttle, and brake inputs. The specific inputs are chosen each simulation step by the driver model, described in section 4.2. The vehicle's dynamics are implemented in the NVidia PhysX engine. This allows the simulator to model the vehicle's dynamics and communicate relevant features such as slipping as needed by the driving algorithm.

4.1.2 Perception

The perception module provides the interface to gather and store information about the vehicle's surroundings. The basic sensing module in AutonoVi-Sim employs a ray-cast with configurable uncertainty, detection time, classification error rate, and sensor angle / range. This module is sufficient to test scenarios such as late detection or misclassification of pedestrians with minimal intervention. A vehicle can be equipped with multiple sensors with varying angles and fidelity. This allows the vehicle to equip high-fidelity

Figure 3. **Simulated Camera Outputs in AutonoVi-Sim:** Camera data captured from different cameras while (**top row**) passing a pedestrian in the road and (**bottom row**) passing a cyclist. **(A)**: A high-resolution camera. **(B)**: Low-resolution camera. **(C)**: A black-and-white camera. **(D)**: A fish-eye 150 degree view camera.

sensors in the longitudinal directions and broader, less accurate sensors in lateral directions. In addition, the perception module specifies how the sensors interact with environmental conditions, including performance impacts and uncertainty caused by weather effects.

The perception module provides interfaces to a generic camera interface and Monte-Carlo scanning ray-casts to simulate various sensor types. These interfaces can be extended to implement LIDAR or camera-based neural network classifiers in simulation. The LIDAR can be configured to change the scanning range, angle, and resolution. Similarly, the camera resolution, color parameters, and refresh rate can be configured for each camera sensor. Figure 2 shows an example of sensor configuration and laser scanner in AutonoVi-Sim. Figure 3 demonstrated varying camera setups in AutonoVi-Sim.

4.1.3 Modelling Real Sensors

A core challenge to generating effective perception data is the capacity to replicate the parameters of sensors found on typical autonomous vehicles. Each sensing modality presents unique refresh rates, error rates, and interactions with the vehicle platform itself. The specific data format of a particular camera or LIDAR must be modelled, and the latency expected of the physical sensor must accounted for. We believe AutonoVi-Sim can provide a platform for exploring these issues.

The sensor system in AutonoVi-Sim is modular, and allows for specifying parameters of the sensors. A sensor's refresh rate can be configured independently of the vehicle, and the expected and output data formats are configurable for sensors of different types. In addition, cameras can be configured for color range, focal length, and camera intrinsic such focal length and distortion, and they can be modeled under varying noise conditions.

The perception systems built on top of imperfect sensor systems can be modelled as well. Misclassification is a typical problem in vision in which an object is assigned an incorrect category. For example, a pedestrian could be misclassified as debris in the road, or a vehicle misidentified as part of the background. By exploiting simulated data from imperfect cameras, we can model classification error and observe and correct the relevant features which cause the vehicle to misidentify and respond inappropriately to nearby entities.

4.2. Drivers

Driving decisions in AutonoVi-Sim, including routing and control inputs, are made by driver models. A driver model fuses information from the road network and the vehicle's sensors to make appropriate decisions for the vehicle. The specific update rate of the driver model can be configured as well as what sensors the model supports and prefers. Each model can implement any necessary parameters needed for the specific approach.

AutonoVi-Sim currently implements three driver models. The **Basic Driver** is a simple lane-following approach which employs control methods similar to a driver assistance lane-keeping system. This driver model is used to generate passive vehicles travelling along their destinations without aberrant or egocentric behaviors. These vehicles are capable of lane-changes and turns, but follow simple rules for these maneuvers and rely on perfect sensing models to accomplish them.

At each planning step, the Basic Driver projects the positions of nearby entities into the future by a pre-determined time threshold. It leverages these projections to exclude choices of control inputs which would lead to a collision with its neighbors. It then chooses the closest speed to its target speed that avoids potential collisions.

We have implemented a more extensive driving model

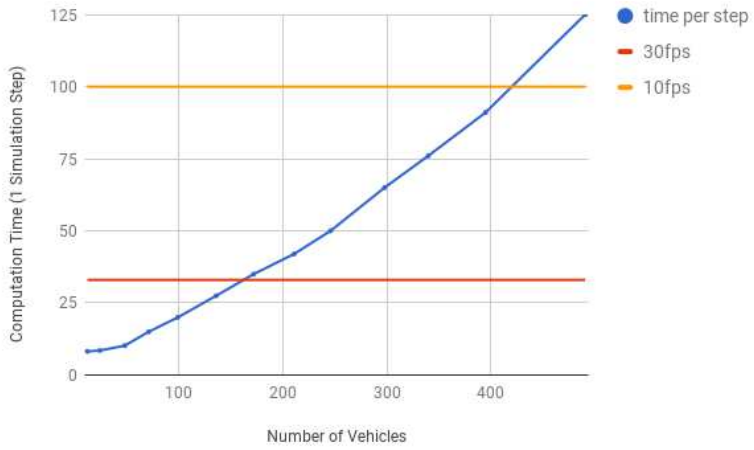


Figure 4. AutonoVi-Sim Performance Graph: We conducted repeated simulations on a traffic network, increasing the number of vehicles in each scenario. This graph details the computation time for each simulation step as a function of the number of vehicles simulated. The limit of 30 frames per second and 10 frames per second are shown for reference. In this scenario, each vehicle is equipped with two basic ray-cast sensors with perfect accuracy. We find that the computation time scales linearly in the number of vehicles simulated, with the ability to simulate 160 vehicles at 30 frames per second and up to 420 vehicles at 10 frames per second.

originally described in [5], the **AutonoVi Driver**. This model uses optimization-based maneuvering with traffic constraints to generate behaviors such as overtaking and combines steering and braking maneuvers through a data-driven vehicle dynamics prediction model. At each planning-step, the model uses a modified control-obstacle [4] formulation to avoid collisions and determines the best control for the next step using a sampling-based approximation of a multi-objective optimization function.

Finally, the simulator implements a **Manual Driver**, which can be activated from any autonomous driver. Manual mode allows an engineer to drive the vehicle using a keyboard, game-pad, or steering wheel and pedal combination. The authors of [3] demonstrate using this manual operation to test vehicle signalling and connected vehicle operation. It can also be used to collect data for neural-network methods, as shown in figure 3.

5. Results

In this section, we provide an overview of several scenarios we have tested in AutonoVi-Sim, training data we have generated, and provide performance results for large-scale traffic simulations. Our results were gathered on a desktop PC running Windows 10, with a quad-core Intel Xeon processor, NVIDIA TitanX gpu, and 16 gb ram.

5.1. Performance Timing

We conducted a series of repeated traffic trials to determine the expected performance of AutonoVi-Sim. We find that the computational costs scale approximately linearly with the number of vehicles simulated. We have successfully simulated over 400 vehicles simultaneously at high-densities at interactive simulation rates. Figure 4 shows the results of our performance tests.

5.2. Autonomous Driving Scenarios

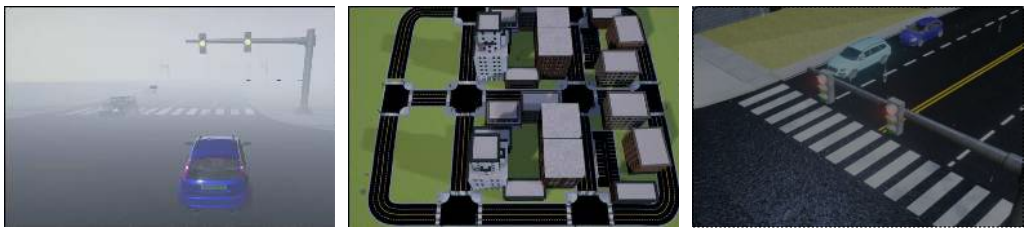
We have implemented the behavior benchmarks described in [5] in Autonovi-Sim to test an autonomous vehicle under challenging conditions. In these benchmarks, the vehicle under observation is referred to as the ego-vehicle.

Passing a bicycle: the ego-vehicle must pass a bicycle on a four-lane road. The vehicle should maintain a safe distance from the bicycle, changing lanes if possible to avoid the cyclist. This scenario can be configured for the density of surrounding traffic to prevent the vehicle from passing without adjusting its speed.

Jaywalking Pedestrian: The vehicle must react quickly to safely decelerate or stop to avoid a pedestrian stepping into the road in front of the vehicle.

Sudden Stop at High Speed: The ego-vehicle must execute an emergency stop on a highway at high speeds when the vehicle in front of it stops suddenly. Autonovi-Sim supports configuring the density and location of surrounding traffic. This allows engineers to test the vehicle in conditions where swerving is not executed simply and must account for surrounding traffic.

High-Density Traffic Approaching a Turn: The ego-vehicle approaches a stoplight at which it must execute a turn, but the ego-vehicle's lane is congested by slow traffic. To make optimal progress, the ego-vehicle should execute a



(A) (B) (C)

Figure 5. **Simulated scenarios and conditions in Autonovi-Sim: (A)**: Heavy fog obstructs the view of the vehicle. **(B)**: 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. **(C)**: Vehicles pass through a slick intersection during rainy conditions.

lane change to the adjoining lane and return to the correct lane with sufficient time to execute the turn.

Car Suddenly entering Roadway: The ego-vehicle travels along a straight road at constant speed when a vehicle suddenly enters the roadway, blocking the ego-vehicle's path. The ego-vehicle must decelerate and swerve to avoid colliding with the blocking vehicle. The speed of the ego-vehicle is configurable. Consistent with [5], we test the ego-vehicle at 10, 30, and 50 mph and with the blocking vehicle obstructing either the right lane or both lanes.

S-turns: The ego-vehicle navigating a set of tight alternating turns, or S turns.

Simulated City: The ego-vehicle navigates to several key points in a small simulated city. The vehicle must execute lane changes to perform various turns as it obeys traffic laws and navigates to its goals. The vehicle encounters bicycles, pedestrians, and other vehicles as it navigates to its goal.

Figure 6 and figure 5 demonstrate several additional scenarios and configurations we have tested in AutonoVi-Sim.

5.3. Generating Training Data

AutonoVi-Sim can be used to generate labelled training data for typical as well as atypical and dangerous situations. We can simulate many scenarios involving pedestrians, cyclists, and other vehicles, such as jaywalking or passing in traffic [5]. The vehicle can be driven automatically using the driver models, or manually by an engineer. Camera, LIDAR, relative position, detection, and control data are exported from each trial of the simulation. The controls of the vehicle combined with local conditions can be used for reinforcement learning in the autonomous driving case or imitation learning in the manual case. These scenarios can be repeatedly run under varying lighting and weather conditions; different surroundings, i.e. buildings, trees, etc; and with different pedestrians, cyclists, and vehicle shapes and sizes. Figure 3 demonstrates a vehicle with 4 co-located cameras of varying properties capturing interactions with a cyclist and pedestrian.

6. Conclusion

We have presented AutonoVi-Sim, a platform for autonomous vehicle simulation with the capacity to represent various vehicles, sensor configurations, and traffic conditions. We have demonstrated AutonoVi-Sim's applicability to a number of challenging autonomous-driving situations and detailed the ways in which AutonoVi-Sim can be used to generate data for training autonomous-driving approaches. AutonoVi-Sim is a modular, extensible framework. While many modules currently represent preliminary implementations of advanced functionality, the extensible nature of the framework provides the basis for progress in the various disciplines which define autonomous driving.

Our work is in active development and still faces a number of limitations. AutonoVi-Sim contains basic implementations of the various modules such as sensors for perception, a physics engine to simulate dynamics etc. However, each of these modules can be extended to more accurately reflect real world conditions. For example, our sensor models currently do not model noise or uncertainty in the exported data. The basic driver behavior is also quite limited; in the future we intend to model additional driver models to provide more rich behaviors for other vehicles.

AutonoVi-Sim currently lacks calibration information to replicate specific sensors and sensor configurations. In the future we hope to model specific sensing packages and algorithms to test specific real-world configurations. In addition, it will be beneficial to explore the transfer between algorithms trained on AutonoVi-Sim and actual test vehicles. Our current driver models are limited to hierarchical, rule-based driving approaches. In future work, we intend to include exploration of end-to-end approaches, which can be represented by a novel Driver model. The current simulator supports few hundreds of vehicles. By combining our simulator with macroscopic or hybrid traffic simulation approaches, we seek to increase the size of supported traffic conditions.

(A) (B) (C)

Figure 6. **Simulated Reactive Scenarios in AutonoVi-Sim: (A)**: The vehicle must stop as a pedestrian enters the roadway suddently. **(B)**: The vehicle changes lanes to pass a cyclist on a four lane road. **(C)**: The basic driver model provides expected positions and projections visually while stopping at a stop-light. AutonoVi-Sim provides interfaces for modelling strategy specific visual debugging information.

7. Acknowledgement

This research is supported in part by Intel, and the Florida Department of Transportation (FDOT) under contract number BDV24-934-01. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

References

[1] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen. Experience, Results and Lessons Learned from Automated Driving on Germany's Highways. *IEEE Intelligent Transportation Systems Magazine*, 7(1):42–57, 2015. 2

[2] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma. Design and development of an optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *Springer Tracts in Advanced Robotics*, 70(STAR):39–54, 2011. 2

[3] D. Barber and A. Best. Connected and automated vehicle simulation to enhance vehicle message delivery. In *8rd International Conference on Applied Human Factors and Ergonomics AHFE, Los Angeles, USA*, In Press. 6

[4] D. Bareiss and J. van den Berg. Generalized reciprocal collision avoidance. *The International Journal of Robotics Research*, 34(12):1501–1514, oct 2015. 3, 6

[5] A. Best, S. Narang, D. Barber, and D. Manocha. Autonovi: Autonomous vehicle planning with dynamic maneuvers and traffic constraints. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, In Press. 4, 6, 7

[6] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016. 1

[7] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat. MPC-based approach to active steering for autonomous vehicle systems. *International Journal of Vehicle Autonomous Systems*, 3(2/3/4):265, 2005. 2

[8] B. Chen and H. H. Cheng. A review of the applications of agent technology in traffic and transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):485–497, 2010. 2

[9] S. Curtis, A. Best, and D. Manocha. Menge: A modular framework for simulating crowd movement. *Collective Dynamics*, 1(0):1–40, 2016. 3

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning (CoRL)*, 2017. 2

[11] K. et al. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009. 2

[12] P. et al. Perception, Planning, Control, and Coordination for Autonomous Vehicles. *Machines*, 5(1):6, 2017. 2

[13] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson. Multipolicy Decision-Making for Autonomous Driving via Changepoint-based Behavior Prediction. *Robotics: Science and Systems*, 2015. 1, 2

[14] E. Galceran, R. M. Eustice, and E. Olson. Toward Integrated Motion Planning and Control using Potential Fields and Torque-based Steering Actuation for Autonomous Driving. *IEEE Intelligent Vehicles Symposium*, (Iv), 2015. 2

[15] J. Hardy and M. Campbell. Contingency Planning Over Probabilistic Obstacle Predictions for Autonomous Road Vehicles. *IEEE Transactions on Robotics*, 29(4):913–929, aug 2013. 2

[16] P. Hidas. Modelling vehicle interactions in microscopic simulation of merging and weaving. *Transportation Research Part C: Emerging Technologies*, 13(1):37–62, 2005. 2

[17] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *IEEE International Conference on Robotics and Automation*, pages 1–8, 2017. 2

[18] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015. 2

[19] S. Kolski, D. Ferguson, M. Bellino, and R. Siegwart. Autonomous Driving in Structured and Unstructured Environments. In *2006 IEEE Intelligent Vehicles Symposium*, pages 558–563. IEEE, 2006. 2

[20] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, NY, USA, 2006. 2

[21] M. Likhachev and D. Ferguson. Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles. *The Interna-*

tional Journal of Robotics Research, 28(8):933–945, 2009. 2

[22] D. L. Margolis and J. Asgari. Multipurpose models of vehicle dynamics for controller design. In *SAE Technical Paper*. SAE International, 09 1991. 2

[23] C. C. T. Mendes, V. Frémont, and D. F. Wolf. Exploiting fully convolutional neural networks for fast road detection. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3174–3179. IEEE, 2016. 1

[24] J. Reeds and L. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990. 2

[25] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. *Playing for Data: Ground Truth from Computer Games*, pages 102–118. Springer International Publishing, Cham, 2016. 2

[26] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan. Planning for Autonomous Cars that Leverage Effects on Human Actions. *Proceedings of Robotics: Science and Systems*, 2016. 2

[27] M. Saifuzzaman and Z. Zheng. Incorporating human-factors in car-following models: A review of recent developments and research needs. *Transportation Research Part C: Emerging Technologies*, 48:379–403, 2014. 2

[28] H. Sun, W. Deng, S. Zhang, S. Wang, and Y. Zhang. Trajectory planning for vehicle autonomous driving with uncertainties. *ICCSS 2014 - Proceedings: 2014 International Conference on Informative and Cybernetics for Computational Social Systems*, pages 34–38, 2014. 2

[29] M. Treiber, A. Kesting, and D. Helbing. Delays, inaccuracies and anticipation in microscopic traffic models. *Physica A: Statistical Mechanics and its Applications*, 360(1):71–88, 2006. 2

[30] J. Tumova, G. C. Hall, S. Karaman, E. Frazzoli, and D. Rus. Least-violating control strategy synthesis with safety rules. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 1–10. ACM, 2013. 2

[31] J. Van Den Berg, J. Snape, S. J. Guy, and D. Manocha. Reciprocal collision avoidance with acceleration-velocity obstacles. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3475–3482, 2011. 2

[32] J. Ziegler, P. Bender, T. Dang, and C. Stiller. Trajectory planning for Bertha - A local, continuous method. *The International Journal of Robotics Research*, 35(April):450–457, 2014. 2