AUTONOVI-SIM: MODULAR AUTONOMOUS VEHICLE SIMULATION PLATFORM SUPPORTING DIVERSE VEHICLE MODELS, SENSOR CONFIGURATION, AND TRAFFIC CONDITIONS

Andrew Best, Corresponding Author
University of North Carolina Chapel Hill
Department of Computer Science
201 S. Columbia St, UNC-Chapel Hill, Chapel Hill, NC 27599-3175
Tel: 919-590-6105 Email: best@cs.unc.edu

Sahil Narang
University of North Carolina Chapel Hill
Department of Computer Science
201 S. Columbia St, UNC-Chapel Hill, Chapel Hill, NC 27599-3175
Tel: 919-590-6105 Email: sahil@cs.unc.edu

Lucas Pasqualin
University of Central Florida
Institute for Simulation and Training
Partnership 2 and 3 Buildings, 3100 Technology Pkwy, Orlando, FL 32826
Tel: 407-882-1300 Email: Lucas.pasqualin@knights.ucf.edu

Daniel Barber
University of Central Florida
Institute for Simulation and Training
Partnership 2 and 3 Buildings, 3100 Technology pkwy, Orlando, FL 32826
Tel: 407-882-1300 Email: dbarber@ist.ucf.edu

Dinesh Manocha
University of North Carolina Chapel Hill
Department of Computer Science
201 S. Columbia St, UNC-Chapel Hill, Chapel Hill, NC 27599-3175
Tel: 919-590-6105 Email: dm@cs.unc.edu

Word Count: 5652 words + 4 figure(s) x 250 + 0 table(s) x 250 = 6652 words

Submission Date: August 1, 2017

ABSTRACT

This paper introduces AutonoVi-Sim, a novel high-fidelity simulation platform for testing autonomous driving algorithms. AutonoVi-Sim is a collection of high-level extensible modules which allows for the rapid development and testing of vehicle configurations, and facilitates construction of complex road networks. Autonovi-Sim supports multiple vehicles with unique steering or acceleration limits, as well as unique tire parameters and overall vehicle dynamics profiles. Engineers can specify the specific vehicle sensor systems and vary time of day and weather conditions to 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 also facilitates data analysis, allowing for capturing video from the vehicle's perspective, exporting sensor data such as relative positions of other traffic participants, camera data for a specific sensor, 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.

Keywords: Autonomous Driving, Self-driving Cars, Traffic Simulation

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 is the safety considerations 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 these 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 condition 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.

The ability to test a driving algorithm in a high-fidelity, physics driven simulation would allow for testing novel approaches without incurring intense labor costs. New vehicles or novel sensor configurations could be explored on many scenarios at once under a wide array of traffic and weather conditions. Engineers could also test the driving algorithm under conditions which may be too dangerous for a real vehicle. For example, engineers could test how the vehicle would respond to erratic behavior from other drivers, pedestrians, or cyclists without endangering the vehicle's passengers or other traffic participants. Insights gained from simulation would provide critical information on algorithmic inefficiencies before actual vehicle testing.

In an effort to provide such a testing platform, this paper introduces AutonoVi-Sim, a simulation framework for 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:

- **Varying vehicle types and traffic density**: AutonoVi-Sim includes various vehicle models allowing for training sensors on differing vehicle shapes, sizes, and colors. In addition, AutonoVi-Sim provides high fidelity traffic simulation, supporting dynamic changes in traffic density and the capacity to model the diversity and dynamics of surrounding vehicles.
- **Rapid Scenario Construction**: Typical road networks can be easily laid out using spline painting and are automatically connected into searchable graphs. This enables the rapid generation of complex, and realistic road networks which support routing and navigation automatically. AutonoVi-Sim supports many lane configurations and atypical road geometry such as cloverleaf overpasses.
- **Cyclists and Pedestrians**: Non-vehicle traffic can be included in a scenario as part of the larger simulation or given specific scripted parameters. Cyclists and pedestrians are supported and can be given navigation destinations like vehicles or specific scenario behaviors to test the ego-vehicle's ressponse time (e.g. walking intro the road in front of the ego-vehicle).
- **Varied Sensor Configurations**: Sensor placement can be varied per-vehicle to determine

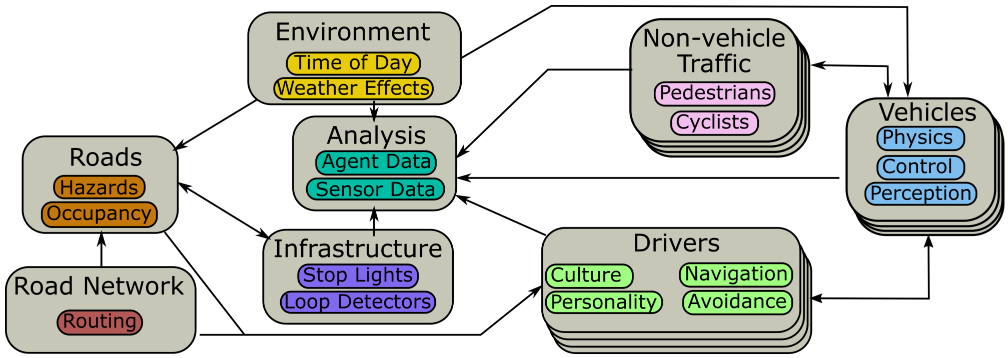

FIGURE 1 The eight modules composing AutonoVi-Sim encompass varying aspects of autonomous driving. The Road, Road Network, and Infrastructure modules define the driving environment. The Environment module allows engineers to specify specific environment conditions including time of day and weather. The 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.

how a particular approach responds to differing environmental information. At runtime, sensor failure or loss of fidelity can be simulated as well.

The rest of the paper is organized as follows. "Related Work" motivates simulation as a tool for advancing autonomous driving and detailed related work in the field. "Simulation Modules" details the core modules provided by AutonoVi-Sim. "Autonomous Driving Modules" discusses the two vehicle related modules which form the core of the autonomous driving component. Results and demonstrative scenarios are given in "Results".

RELATED WORK

Engineers in numerous disciples have focused on simulating vehicles and traffic for research and commercial purposes. MATSim (*1*) provides the macroscopic simulation of large-scale traffic networks of thousands of vehicles. SUMO (*2*) offers modular agent-based simulation of traffic networks from a 2D perspective. Approaches to modelling hybrid networks of agents and macroscopic flow have been proposed (*3*). In these simulations, the underlying physics and dynamics of the vehicles are coarsely approximated or neglected as the traffic flow patterns are the principle focus of the studies. However, for autonomous driving approaches, high-fidelity physics driven simulations are necessary to train algorithms.

Simulation has been an integral tool in the development of controllers for autonomous vehicles. (*4*), (*5*), and (*6*) 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 (*7–9*). Recent work in traffic modelling has sought to increase the fidelity of the modelled drivers and vehicles (*10*).

Recent studies support the use of high-fidelity microscopic simulation for data-gathering and training of vision systems. (*11*) and (*12*) 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(*13*) platform. Using video game data provides access to relatively high-fidelity vehicle models but limits the ability to implement sensing systems and access data beyond visual data. A 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.

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 (*14*). 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 (*15*). The Reeds-Shepp formulation is a widely used car model with forward and backward gears (*16*). Margolis and Asgari (*17*) present several representations of a car including the widely used single-track bicycle model. Borrelli et al. (*15*) extend this model by including detailed tire-forces.

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 (*18*) and data-driven models such as (*19*). Logic-based approaches with safety guarantees have also been demonstrated (*20*). An extensive discussion on techniques to model these behaviors in traffic simulation can be found in (*10*).

Path Planning and Collision Avoidance: Prior approaches to path planning for autonomous vehicles are based on random-exploration (*21*), occupancy grids (*22*), potential-field methods (*23*), driving corridors (*24*), etc. Recent approaches seek to incorporate driver behavior prediction in path planning using Bayesian behavior modeling (*25*) and game-theoretic approaches (*26*). Continuous approaches for collision-avoidance have been proposed based on spatial decomposition or velocity-space reasoning. Ziegler et al. (*27*) 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. (*29*) apply velocity-space reasoning with acceleration constraints to generate safe and collision-free velocities. Bareiss et al. (*30*) extend the concept of velocity obstacles into the control space to generate a complete set of collision-free control inputs.

SIMULATION MODULES

Drawing from recent work in crowd simulation, (*31*), 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. This section details the modules which make up the basic simulation system, reserving discussion of the vehicle and driving strategy modules for the subsequent section.

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 features for modelling specific environments. FIGURE 3(A) shows an example of road placement in AutonoVi-Sim.

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 the Vehicles module below. Infrastructure components are provided with basic detection capability. For example, traffic lights can determine which lanes are congested and adjust the light cycle accordingly to more traffic more effectively.

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 the roads into a directed graph. The Road Network provides GPS style routing to vehicles and localization for mapping purposes. Coupled with the Roads 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.

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. Environmental effects such as road friction reduction are controlled by the environment module.

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 a target 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 a target vehicle approaches, a cyclist can be

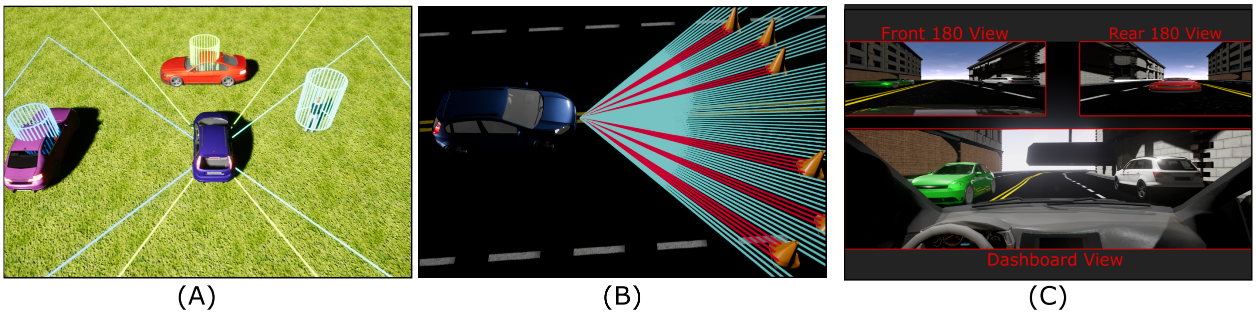

FIGURE 2 This figure shows examples of AutonoVi-Sim sensor setup. (A): The ray-cast sensors provide a simple interface for sensing nearby objects. Each sensor is shown in its own color, has a unique range, and a unique detection uncertainty value. (B): An example configuration of a hatchback with a laser rangefinder navigating around traffic cones. Reflected beams are illustrated in red. Beams which do not return hits are illustrated in cyan for debugging. (C): An example camera configuration for a test vehicle. A 180 degree forward facing camera, a 180 degree rear-facing camera, and a dashboard camera are illustrated.

triggered to stop in the road, suddenly change direction, or enter the road in an unsafe fashion.

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. Specific sub-routines of the vehicle can be timed, profiled, and logged using a simple interface. Camera-based sensors can record out the video data captured during simulation as can LIDAR based sensors. The Vehicles section describes sensors in more detail.

AUTONOMOUS DRIVING MODULES

The simulation modules described in the prior section 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 i.e. the Vehicles and Drivers modules.

Vehicles

A 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 even vary at runtime as needed.

Control and Dynamics

Vehicle control is provided via three axes: steering, throttle, and brake inputs. The specific inputs are determined at each simulation step by the driver model, described below. 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.

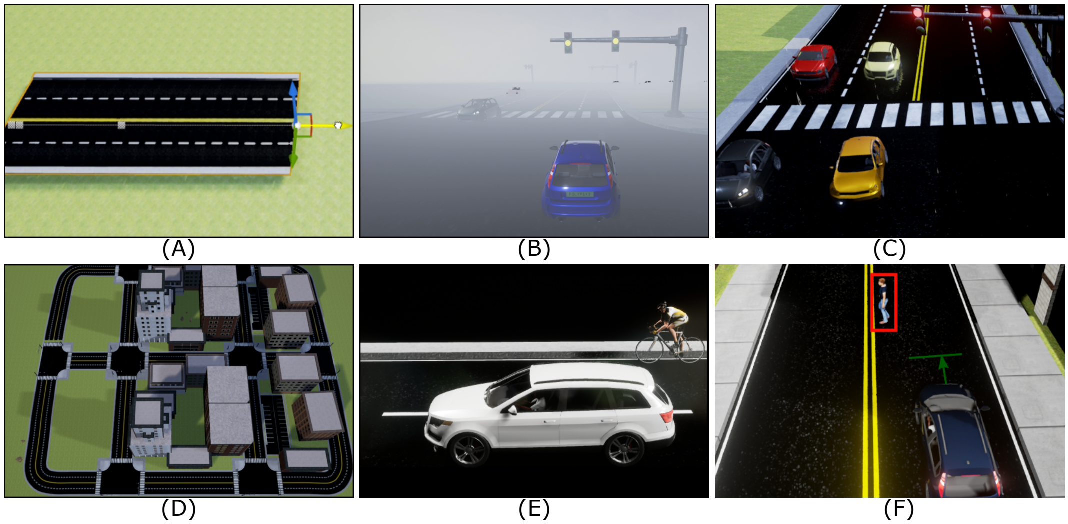

FIGURE 3 (A): Road networks in Autonovi-Sim are easily constructed by dragging splines along the landscape. The roads automatically connect into a road network. (B): Heavy fog obstructs the view of the vehicle. (C): Vehicles pass through a slick intersection during rainy conditions. (D): Closed-circuit road networks such as this simulated city block allow engineers to test driving algorithms over long timescales by assigning new navigation goals periodically. (E): An SUV navigating with AutonoVi changes lanes and passes a cyclist safely. (F): A hatchback navigating with AutonoVi detects a jaywalking pedestrian and generates a stopping point to wait for the pedestrian to cross.

Perception

The perception module of a vehicle provides the interface for gathering and storing information about the surroundings. The basic sensing module in AutonoVi-Sim employs a ray-cast with configurable detection noise, 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, for example, for the vehicle to simulate high-fidelity sensors in the longitudinal directions and broader, less accurate sensors in lateral directions. In addition, interaction with environmental conditions can be specified for the basic sensors, including performance impacts and uncertainty caused by weather effects. FIGURE 2(A) shows an example of a basic 4-sensor setup on a vehicle.

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(B) and (C) show an example of a camera-based sensor and simple laser rangefinder.

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.

The more extensive driver,the **AutonoVi Driver**, is described in detail in (*32*). 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, AutonoVi uses a modified control-obstacle (*30*) formulation to avoid collisions and determines the best control for the next step using a sampling-based approximation of a multi-objective optimization function.

The simulator also implements the **Manual Driver**. The Manual Driver can be activated from any autonomous driver. It allows an engineer to drive the vehicle using a keyboard, gamepad, or steering wheel and pedal combination. As described in (*33*), this manual operation is being employed to test vehicle signalling and connected vehicle operation. It can also be used to collect data for neural network-based methods.

RESULTS

This section provides an overview of several scenarios that have been tested in AutonoVi-Sim and provide performance results for large-scale traffic simulations. Results have been gathered on a desktop PC running Windows 10, with a quad-core Intel Xeon processor, NVIDIA TitanX gpu, and 16 gb ram. An extended video demonstration of the results can be found at http://gamma.cs.unc.edu/AutonoVi/.

FIGURE 3 demonstrates several environmental effects simulated in AutonoVi-Sim. FIGURE 3(B) demonstrates fog obscuring the view of a vehicle as it navigates an intersection. Fog can be introduced in a scenario to reduce visibility and impede sensor accuracy. The density of fog can be configured dynamically to reflect changing conditions. FIGURE 3(C) demonstrates vehicles navigating wet roads during a rainstorm. The friction of road surfaces decrease as wetness increases. The road material can be specified and the effects of rain on each surface can be configured independently.

Autonomous Driving Scenarios

As part of (*32*), a set of specific behavior benchmarks for Autonovi-Sim was designed to test an autonomous vehicle under challenging conditions. Visual demonstrations and extended details of the scenarios below can be found in (*32*).

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.

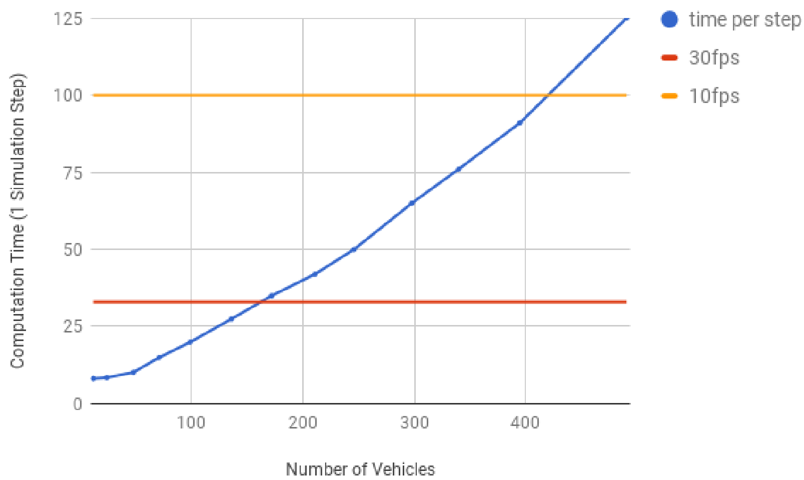

FIGURE 4 To test AutonoVi-Sim's performance scaling, repeated trials were conducted on a road network similar to FIGURE 3(A), with increasing numbers of vehicles in each trial. Each vehicle is equipped with two basic ray-cast sensors with perfect accuracy. This graph details the computation time for each simulation step as a function of the number of vehicles simulated. The limit of 30 frames per second (FPS) and 10 FPS are shown for reference. Results show that the computation time scales approximately linearly in the number of vehicles simulated, with the ability to simulate 160 vehicles at 30 fps and up to 420 vehicles at 10 FPS.

Sudden Stop at High Speed: The 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 of surrounding traffic. This allows for testing the vehicle in conditions where swerving is not executed simply and must account for surrounding traffic.

Passing a cyclist: the vehicle must pass a cyclist on a four-lane road. The vehicle must maintain a safe distance from the cyclist, 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.

Car Suddenly entering Roadway: The vehicle travels along a straight road at constant speed when a vehicle suddenly enters the road from a side street, blocking the vehicle's path. The vehicle must decelerate and swerve to avoid colliding with the blocking vehicle. The speed of the vehicle is configurable. In (*32*), the vehicle was tested at 10, 30, and 50 mph and with the blocking vehicle obstructing either the right lane or both lanes.

S-turns: The vehicle must navigate a set of tight alternating turns, or S turns. The underlying vehicle dynamics yield different speeds and safety parameters for each vehicle type.

High-Density Traffic Approaching a Turn: The vehicle approaches a stoplight at which it must execute a turn, but the current travel lane is congested by slow traffic. To make optimal progress, the vehicle should execute a lane change to the adjoining lane and return to the correct lane with sufficient time to execute the turn.

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

Performance

A series of repeated traffic trials was conducted to determine the expected performance of AutonoVi-Sim. Results indicate that computational costs scale approximately linearly with the number of vehicles simulated. During testing, over 400 vehicles were simulated simultaneously at high-densities at interactive simulation rates. FIGURE 4 shows the results of the performance tests.

CONCLUSION

This paper has presented AutonoVi-Sim, a platform for autonomous vehicle simulation with the capacity to represent various vehicles, sensor configurations, and traffic conditions. It has demonstrated AutonoVi-Sim's applicability to a number of challenging autonomous-driving situations and detailed the ways in which AutonoVi-Sim can enhance the state of the art in testing autonomous-driving algorithms. 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.

This 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, AutonoVi-Sim currently lacks calibration information to replicate specific sensors and sensor configurations.

In future work, the simulator can be improved by modelling 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. Future work will also explore the integration of AutonoVi-Sim with prior approaches to macroscopic traffic simulation to generate hybrid multi-scale traffic simulations similar to (*3*). This would allow the simulation of substantially larger traffic flows. The individual driver models could also be improved by incorporating behavior prediction for nearby traffic.

ACKNOWLEDGEMENTS

This research is supported in part by ARO grant W911NF-16-1-0085, NSF grant 1305286, 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] Horni, A., K. Nagel, and K. W. Axhausen, *The multi-agent transport simulation MATSim*. Ubiquity Press London, 2016.

[2] Krajzewicz, D., G. Hertkorn, C. Rössel, and P. Wagner, SUMO (Simulation of Urban MObility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, 2002, pp. 183–187.

[3] Sewall, J., D. Wilkie, and M. C. Lin, Interactive hybrid simulation of large-scale traffic. In *ACM Transactions on Graphics (TOG)*, ACM, 2011, Vol. 30, p. 135.

[4] et al., P., Perception, Planning, Control, and Coordination for Autonomous Vehicles. *Machines*, Vol. 5, No. 1, 2017, p. 6.

[5] Katrakazas, C., 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*, Vol. 60, 2015, pp. 416–442.

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

[7] Anderson, S. J., 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*, Vol. 70, No. STAR, 2011, pp. 39–54.

[8] Likhachev, M. and D. Ferguson, Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles. *The International Journal of Robotics Research*, Vol. 28, No. 8, 2009, pp. 933–945.

[9] Aeberhard, M., 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*, Vol. 7, No. 1, 2015, pp. 42–57.

[10] Chen, B. and H. H. Cheng, A review of the applications of agent technology in traffic and transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 2, 2010, pp. 485–497.

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

[12] Johnson-Roberson, M., 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*, 2017, pp. 1–8.

[13] *OpenAI Universe*. `https://universe.openai.com/`, 2017, accessed: 2017-07-01.

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

[15] Borrelli, F., 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*, Vol. 3, No. 2/3/4, 2005, p. 265.

[16] Reeds, J. and L. Shepp, Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, Vol. 145, No. 2, 1990, pp. 367–393.

[17] Margolis, D. L. and J. Asgari, Multipurpose Models of Vehicle Dynamics for Controller Design. In *SAE Technical Paper*, SAE International, 1991.

[18] Treiber, M., A. Kesting, and D. Helbing, Delays, inaccuracies and anticipation in microscopic traffic models. *Physica A: Statistical Mechanics and its Applications*, Vol. 360, No. 1, 2006, pp. 71–88.

[19] Hidas, P., Modelling vehicle interactions in microscopic simulation of merging and weaving. *Transportation Research Part C: Emerging Technologies*, Vol. 13, No. 1, 2005, pp. 37–62.

[20] Tumova, J., 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*, ACM, 2013, pp. 1–10.

[21] et al., K., Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 5, 2009, pp. 1105–1118.

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

[23] Galceran, E., 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*, , No. Iv, 2015.

[24] Hardy, J. and M. Campbell, Contingency Planning Over Probabilistic Obstacle Predictions for Autonomous Road Vehicles. *IEEE Transactions on Robotics*, Vol. 29, No. 4, 2013, pp. 913–929.

[25] Galceran, E., 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.

[26] Sadigh, D., 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.

[27] Ziegler, J., P. Bender, T. Dang, and C. Stiller, Trajectory planning for Bertha - A local, continuous method. *The International Journal of Robotics Research*, Vol. 35, No. April, 2014, pp. 450–457.

[28] Sun, H., 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*, 2014, pp. 34–38.

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

[30] Bareiss, D. and J. van den Berg, Generalized reciprocal collision avoidance. *The International Journal of Robotics Research*, Vol. 34, No. 12, 2015, pp. 1501–1514.

[31] Curtis, S., A. Best, and D. Manocha, Menge: A Modular Framework for Simulating Crowd Movement. *Collective Dynamics*, Vol. 1, No. 0, 2016, pp. 1–40.

[32] Best, A., 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.

[33] Barber, D. 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.