International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 6318-6322

Processing M-based IoT Visualization for Fast Development of Safe, Collision-Free Autonomous Driving

Dongkyu Lee, Jeonghun Cho and Daejin Park*

Postal: 41566, Daehak-ro 80, Buk-gu, Daegu City, School of Electrnoics Engineering, Kyungpook National University, Republic of Korea.
**Corresponding Authors*

ORCID: 0000-0002-5560-873X, Scopus Author ID: 55463943600

Abstract

Autonomous driving for the maximum throughput of traffic is too complex to cover various cases of exceptions that pertain to large-scale cars in an urban space, while still guaranteeing the safety of the self-organizing algorithm. In this paper, we present a simulation-integrated interactive visualization platform to enable the fast development of a safe, self-organizing driving algorithm. By using the proposed platform, developers describe geographic roads and allocate active cars that are equipped with specific autonomous algorithms. The cross-coupled simulation result of the self-organized movement of cars is interactively visualized with traffic on roads to demonstrate the weakness of the embedded autonomous driving algorithm. As a case study, we present a demonstration for 1000 cars on 300 junctions of roads and offer a way to accelerate the development of self-organizing control in autonomous cars.

INTRODUCTION

Autonomous vehicle control (AVC) gives drivers convenience and increases safety while op-erating their vehicles (1). In addition, AVC can be used to provide maximum throughput of urban traffic by controlling self-organizing movement for large-scale cars crowd (2). The com-plex events (3) that occur within the ever-changing environment of roads affects the movement of the individual cars so that the cross-coupled control of AVC-based cars requires complex algorithms that consider the unexceptional environment (4) by using predictive guidance for collision-free driving (5).

The user-defined algorithm must be applied to real-time traffic situations, and it has to be evaluated to provide predictive guidance (6) for increasing traffic throughput. In this pa-per, we introduce a framework to effectively visualize the safety-conscious performance of the collision-free AVC algorithm for cars on road networks, which is allocated to geographic layers in order to consider the traffic situations (7). The road network is described with the coupled relationship of road segments (8). The traffic information, which is gathered from a public transformation data set, is merged into the road network layer. The road segment consists of multiple tiles that contain fundamental information about geo-graphic road architecture, intersections, traffic, and the autonomous movement of active cars on the street. We adopted a processingT M -based visualization language to decrease complexity in describing the architectural structure on an event-driven simulation kernel. An AVC algorithm is embedded into the independent models of each car and is used to determine the traffic flow of the self-organizing movement of multiple cars, which are dynamically directed by the results of event-driven simulation (9). The proposed platform visualizes these three layers of road, traffic, and car movement simultaneously, so it provides a way to efficiently evaluate the performance of the applied AVC algorithm (10).

PROPOSED ARCHITECTURE

Multi-layered Model Construction

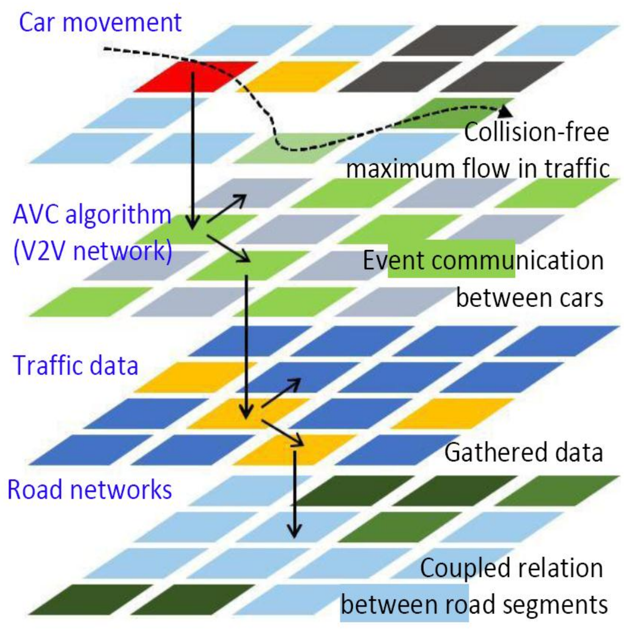

Figure 1: Multi-layered construction of road segments, traffic data, AVC algorithm, and car movement representation

During the process of driving autonomously, there are two types of AVC algorithms. The first type interacts with the surrounding environment for safety purpose (11). It reacts quickly to situations in order to address emergencies. However, it cannot afford the best throughput because it only cares about itself. The second type of AVC algorithm communicates with the network that is connected to each vehicle (12) and the sensors on the road and produces max-imum throughput. Through the network, this algorithm can ascertain the situation on all roads and predict the situation on the route that the car is traveling. As a result, the algorithm guides the cars on the least congested, but not necessarily the most direct route.

When the AVC algorithm controls flow, it needs a number of pieces of information. We develop a multilayered process to separate the algorithm, data and visualizing platform. By classifying data sets according to their purpose, each layer can be created as shown in Fig. 1. The platform consists of several layers, including the car movement layer, the traffic data layer, the AVC algorithm layer and the road network layer. Each layer interacts with each other. In the road network layer, there are road segments that have geographic information. Based on the geographic street in the road network layer, the traffic data layer gathers traffic data, such as the vehicle flow and traffic conditions that are required for the AVC algorithm. The AVC algorithm that needs to be simulated is loaded into the AVC algorithm layer. The AVC algorithm receives the information from each layer and produces the car movement for the given information. The car movement that produced by AVC algorithm is displayed in the car movement layer.

Extensible Road Networks

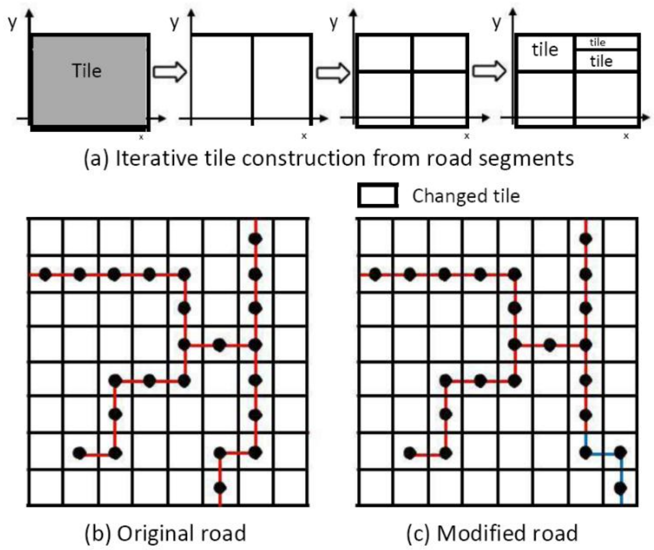

Figure 2: Constructing a tile-based road network

The road network layer is the foundation of all layers. Each small tile consists of all layers, as shown in Fig. 1. In the road network layer, we construct geographic roads into a set of road segment tiles. As in Fig. 2(a), small tiles are gathered together to form large tiles and large tiles are gathered together to form a road network. Using this method that gathers small tiles to construct the road network, the user can effectively modify the road network by changing some tiles rather than reconstructing the entire road network as shown in Fig. 2(b)(c). Also, the user can construct a new road by arranging and inserting basic unit tiles instead of creating a new base tile. Since each road segment is connected to another layer, creating the road network makes it possible to create all of the layers. We create some basic tiles that can make all kinds of road structures and arrange them to make a tile-based road segment. Fig. 3 shows an example of a tile-based road segment construction.

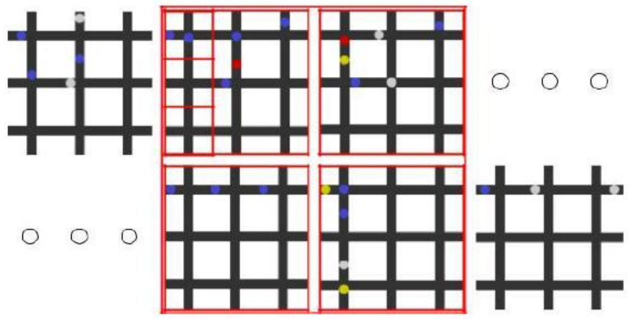

Figure 3: Case study: tile-based road segment construction

Distribution Algorithm and Flow Prediction

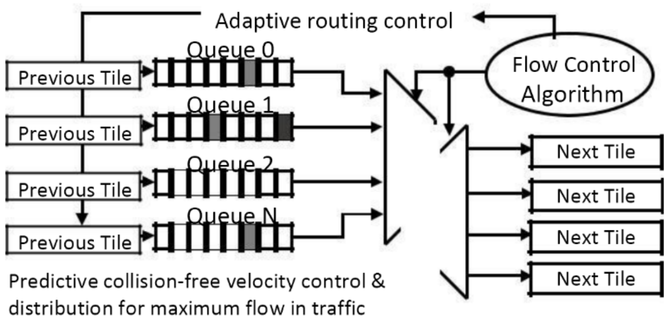

Figure 4: Predictive velocity control and movement distribution of a car for maximizing the vehicular flow in traffic

The traffic data layer is dependent on the road network layer. Traffic data layer collects the data from the road in the road network layer. Using the collected traffic data, the speed of cars and their direction at an intersection are determined by the AVC algorithm. Fig. 4 illustrates the behavior of a platform based on the flow control. There is a queue at each intersection. When the car enters the intersection, the queue piles up. If the car flow comes from the previous tile to the present intersection, the AVC algorithm that was applied in this platform distributes the car flow so that the throughput is maximized according to the states of the following queue. According to Fig. 5, if the car goes from PE to D, the AVC algorithm checks the traffic data on the path that the car wants to take and determines where to go. After the cars arrive at the next intersection, the AVC algorithm repeats the process by level 2. Until the car arrives at its destination, the AVC algorithm repeats the process by increasing the level.

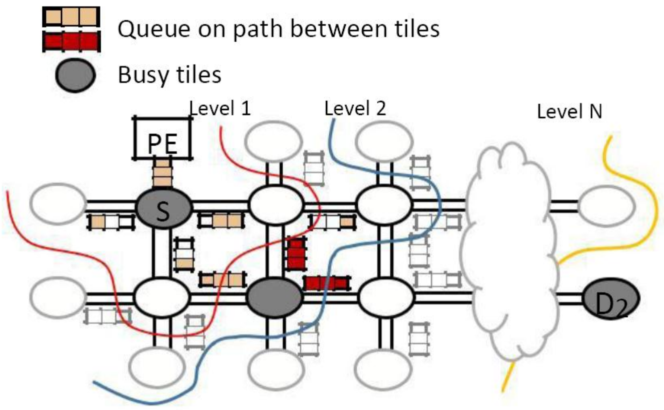

Figure 5: Distribution algorithm in car traffic

In Fig. 6(a)(b), if the algorithm under-predict the distance between the vehicles, the basic speed of the cars will increase. This process will improve the throughput. However, it will increase the probability that a collision will occur. On the contrary, if the distance between the vehicles is over-predicted, then the basic speed will decrease; as a result, both the probability of that a collision will occur and the throughput will decline. Also, changing the acceleration and deceleration will affect the throughput and safety. After the AVC algorithm determines the speed and direction of each car, the platform predicts the position of the vehicles and updates the propagating events into adjacent tiles as in Fig. 6(c)(d). The algorithm moves the flow of the vehicle in the direction in which the throughput is maximized.

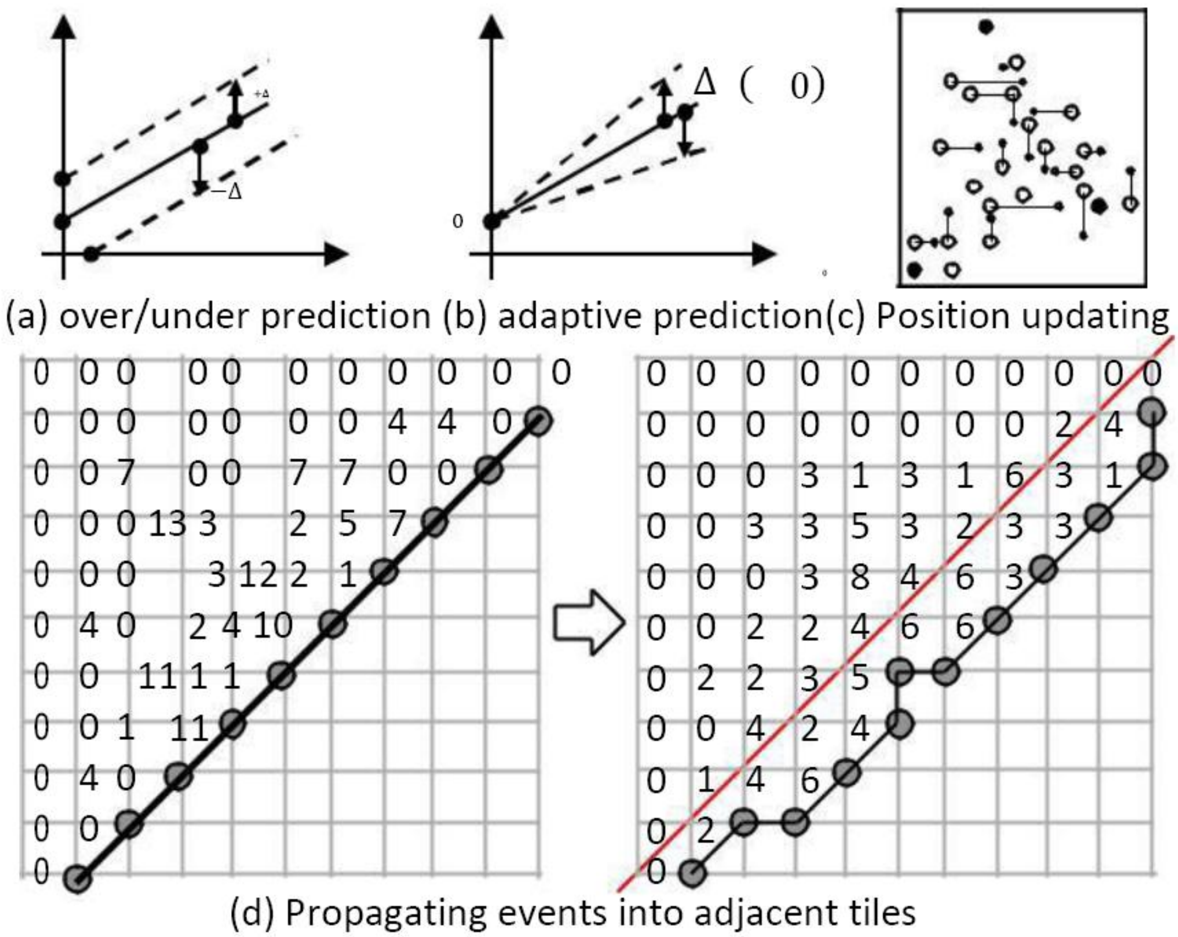

Figure 6: Traffic congestion matrix and collision-free adaptive control of velocity for flow pre-diction

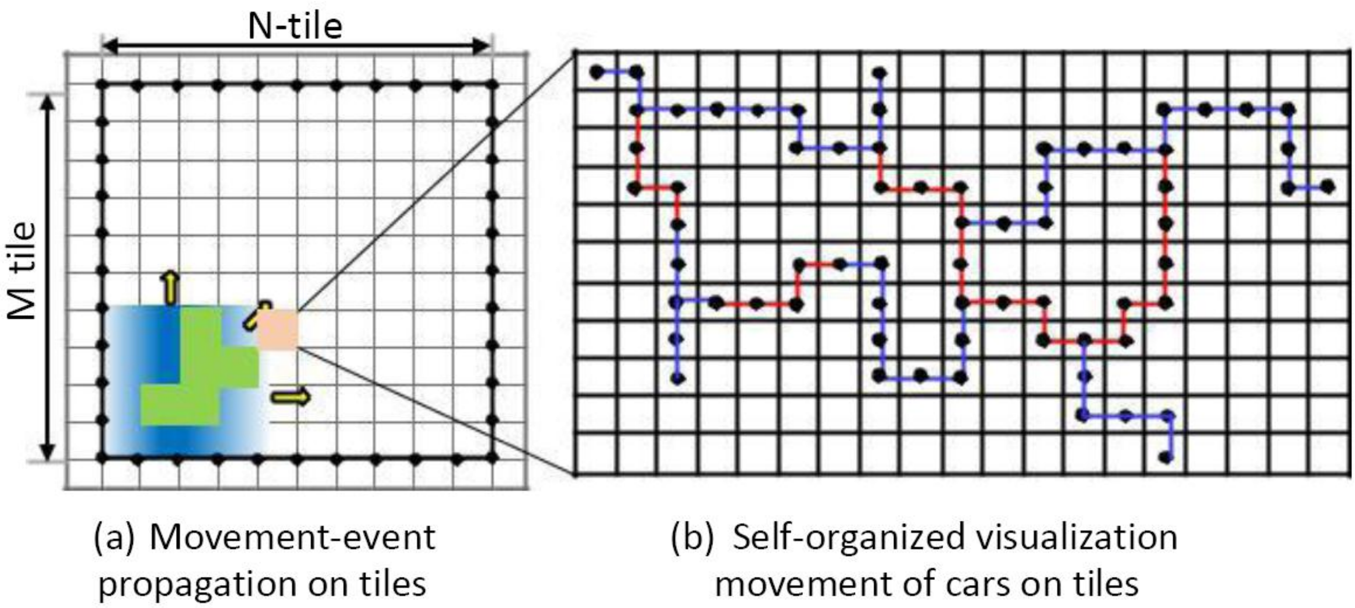

Figure 7: Interactive visualization of the simulation result for cross-layered models and predictive guidance

After the AVC algorithm layer produces car movements, the car movement layer displays the car movement in real time to show how the AVC algorithm that the user applies will react. By abstracting Fig. 6(d), the scenario becomes Fig. 7(a). Each tile that the set of the small tiles visualizes the quantity of the vehicle movement using color. Marking the quantity of the vehicle movements by color intuitively visualizes the flow of traffic, thus making it easier to see the car flow. According to Fig. 7(c), the car movement layer of small tiles displays the car movement that is predicted by the AVC algorithm.

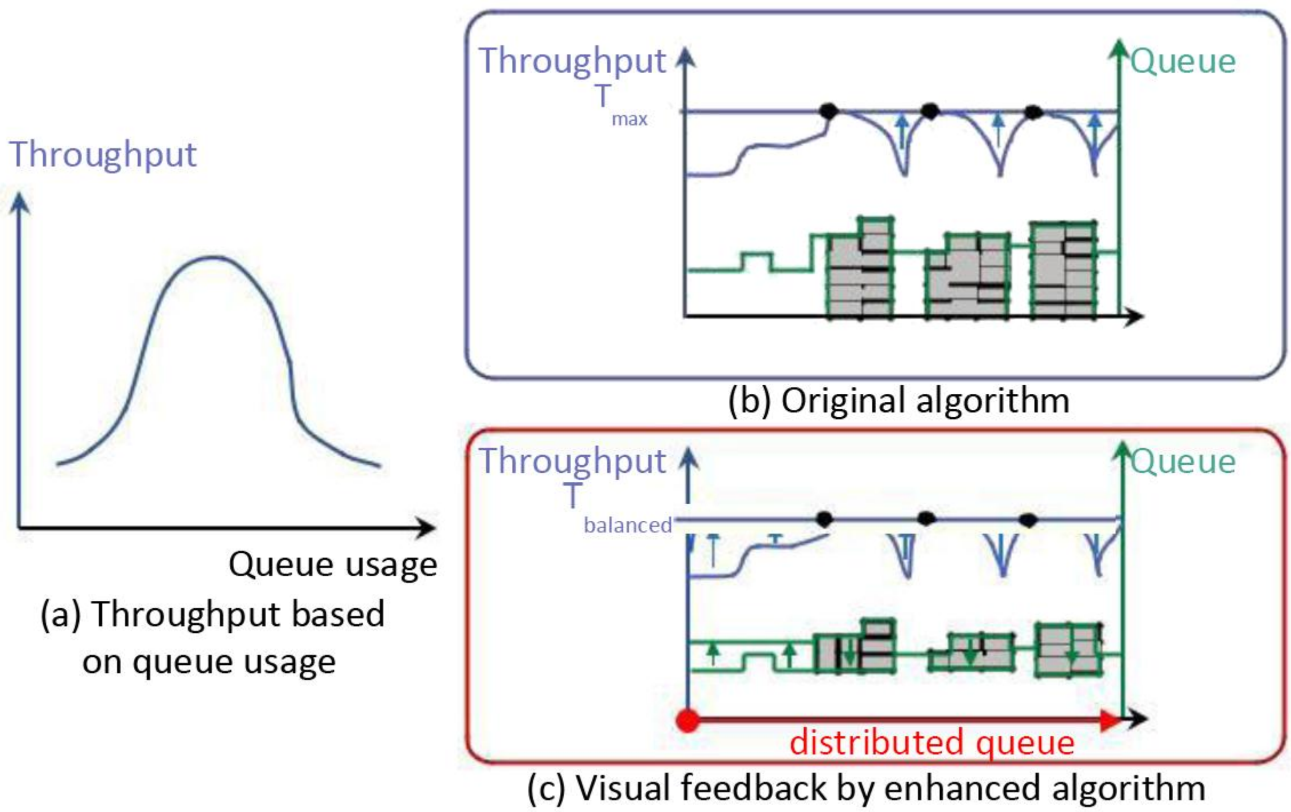

Figure 8: Evaluation by the queuing and distribution algorithm in car traffic

The total throughput varies according to the queue usage. As in Fig. 8(a), when there are too many cars in the queue, the throughput decreases. Since each intersection has a limited amount of cars that can pass through time. If the queue usage is too small, then the throughput also decreases since there are fewer cars that can pass through the system. In order to increase the throughput, the queue usage should be maintained appropriately. Fig. 8(b) represents the average throughput and total queue usage on the red line in Fig. 6(d). In order to increase the throughput, out approach assists to distribute the flow of vehicles to decrease the queue usage and control the speed of the vehicles to decrease the queuing speed using the enhanced AVC algorithm.

CONCLUSIONS

This paper introduces an interactive simulation platform by using processing-based visualization to ensure safe, collision-free autonomous driving. There are many variables in the AVC algorithm. We cannot predict all of the variables, yet this platform provides an easy way to anticipate them. It informs the result of the AVC algorithm immediately and visualizes the results of the changes to the variables. In order to increase the performance of the collision-free AVC algorithm, the AVC algorithm is applied in this platform and iterates a simulation of changing variables. It will not only reduce the cost and time involved in developing a safer and more effective AVC algorithm, but will also allow for the visualization of the performance of the AVC algorithm for easy analysis.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

ACKNOWLEDGMENT

This research was supported by the Basic Science Research Program through the National Re-search Foundation of Korea(NRF) funded by the Ministry of Education (2014R1A6A3A04059410 and 2016R1D1A1B03934343). Corresponding author is Daejin Park (boltanut@knu.ac.kr).

REFERENCES AND NOTES

[1] L. Situ, “Electric vehicle development: The past, present, future,” in Power Electronics Systems and Applications, 2009. PESA 2009. 3rd International Conference on, May 2009, pp. 1–3.

[2] J. Xu, D. Deng, U. Demiryurek, C. Shahabi, and M. v. d. Schaar, “Context-aware online spatiotemporal traffic prediction,” in 2014 IEEE International Conference on Data Mining Workshop, Dec 2014, pp. 43–46.

[3] D. Park, M. D. Yin, and J. Cho, “Secure microcontroller with on-chip hierarchical code validator for firmware authentication,” in IT Convergence and Security (ICITCS), 2014 International Conference on, Oct 2014, pp. 1–3.

[4] J. Youn, M. D. Yin, J. Cho, and D. Park, “Steering wheel-based adaptive headlight con-troller with symmetric angle sensor compensator for functional safety requirement,” in 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), Oct 2015, pp. 619–620.

[5] K. D. Kim, “Collision free autonomous ground traffic: A model predictive control ap-proach,” in 2013 ACM/IEEE International Conference on Cyber-Physical Systems (IC-CPS), April 2013, pp. 51–60.

[6] T. Weiskircher, Q. Wang, and B. Ayalew, “Predictive guidance and control framework for (semi-)autonomous vehicles in public traffic,” IEEE Transactions on Control Systems Tech-nology, vol. PP, no. 99, pp. 1–13, 2017.

[7] C. Shahabi, F. Banaei-Kashani, A. Khoshgozaran, L. Nocera, and S. Xing, “Geodec: A framework to effectively visualize and query geospatial data for decision-making,” IEEE MultiMedia, vol. PP, no. 99, pp. 1–1, 2015.

[8] C. McGeoch, “Analyzing algorithms by simulation: variance reduction techniques and simulation speedups,” ACM Comput. Surv., vol. 24, no. 2, pp. 195–212, Jun. 1992. [Online]. Available: http://doi.acm.org/10.1145/130844.130853

[9] M. Seok, T. Kim, C. Choi, and D. Park, “A scalable modeling and simulation environment for chemical gas emergencies,” Computing in Science and Engineering, vol. 18, no. 4, pp. 25–33, 2016, cited By 0.

[10] K. S. Perumalla, R. M. Fujimoto, P. J. Thakare, S. Pande, H. Karimabadi, Y. Omelchenko, and J. Driscoll, “Performance prediction of large-scale parallel discrete event models of physical systems,” in Winter Simulation Conference, 2005, pp. 356–364.

[11] S. Lee, J. Cho, T. G. Kim, and D. Park, “Things-touch system and matlab/simulink in-teroperation environment based on acoustic event sensing for individualized things-human interaction,” Journal of IEMEK, vol. 10, no. 4, pp. 189–198, Aug 2015.

[12] I. Son, K. Han, D. Park, M. D. Yin, and J. Cho, “A study on implementation of ivi applica-tions for connected vehicle using html5,” in IT Convergence and Security (ICITCS), 2014 International Conference on, Oct 2014, pp. 1–4.