17th World Conference on Earthquake Engineering, 17WCEE
Sendai, Japan - September 27th to October 2nd, 2021

HPC ENHANCED AGENT BASED SYSTEM FOR SIMULATING EVACUATION OF LARGE URBAN AREAS

Maddegedatara Lalith (1), Wasuwat Petprakob(2), Muneo Hori(2), Tsuyoshi Ichimura(3), Kohei Fujita(4)

(1) *Assoc. Prof., Earthquake Research Institute, The Univ. of Tokyo, Bunkyo-ku, Tokyo, Japan. Email: lalith@eri.u-tokyo.ac.jp*
(2) *MAQE Bangkok Co. Ltd. Lumphini, Pathumwan, Bangkok, Thailand. Email: wasuwat.p@hotmail.com*
(3) *Director-General, Research Inst. for Value-Added-Information Generation, JAMSTEC, Japan. Email: horimune@jamstec.go.jp*
(3) *Assoc. Prof., Earthquake Research Institute, The Univ. of Tokyo, Bunkyo-ku, Tokyo, Japan. Email: ichimura@eri.u-tokyo.ac.jp*
(4) *Asst. Prof., Earthquake Research Institute, The Univ. of Tokyo, Bunkyo-ku, Tokyo, Japan. Email: fujita@eri.u-tokyo.ac.jp*

Abstract

We present an HPC enhanced Agent Based Model (ABM) developed with the aim of simulating emergency mass evacuations under uncommon conditions caused by natural disasters. Most of the available evacuation simulators use simplified models that are efficient in modeling ordinary conditions. During a major disaster, like a major earthquake followed by a tsunami, combinations of uncommon conditions like non-functioning traffic signals, large number of pedestrians on roads, partially blocked roads due to debris, etc. can be widespread. It is vital to take these uncommon conditions into account since these can significantly influence the evacuation progress. It is difficult to develop simplified model to simulate these rare conditions due to the lack of previous observations, etc. A versatile approach that can model many combinations of these uncommon events is using Agent Based Models (ABMs) in which the software agents with sufficient capabilities to mimic real evacuees interact to reproduce the reality as an emergent phenomenon. Both the development of such ABMs and the development of HPC extensions to meet their high computational cost are challenging. We developed a scalable HPC enhanced agent-based framework capable of high-resolution large scale simulations. This paper presents some details of the developed ABM for simulating emergency evacuation under a non-functioning traffic signal, in which cars and pedestrians interact on roads and junctions. As a demonstrative application of the developed ABM, we present a hypothetical evacuation scenario in which a selected number of slow pedestrians are allowed to use cars during a given time window and traffic signals are assumed to be non-functioning.

Keywords: Agent based modeling, pedestrian-car interaction, non-signalized junctions, mass evacuation, high performance computing

1. Introduction

Emergency mass evacuations triggered by major disasters, like the impending Tokai, Tonankai and Nankai earthquakes and the associated tsunamis, can produce a wide range of uncommon scenarios like non-functioning traffic signals, debris scattered roads, pedestrians on roads, low lighting, etc. It is vital to consider these uncommon scenarios since those can seriously hamper the progress of evacuation. Most of the existing evacuation simulators are based on simple queue models in 1D networks. These simple models have advantages like the ease of model development, ease of use, low computational demand, etc. Although the simple models can capture common scenarios like traffic or pedestrian congestions reasonably well, they are either ineffective or incapable of modeling uncommon scenarios like pedestrians and cars at a junction with broken traffic signals during a disaster. A versatile approach which can model any combinations of these rare possibilities is using Agent Based Models (ABMs). Once software agents with sufficient capabilities to mimic real evacuees' behaviors are developed, ABMs can be utilized to model different combinations of these uncommon scenarios as an emergent phenomenon of the agents' interactions according to the specified rules defining the agents' behaviors. While developing logic to mimic evacuees' behaviors itself is challenging, the need of scalable high-performance computer implementations to meet the computational demand of a complex ABM significantly increases the required efforts.

With the aim of simulating such uncommon scenarios, we developed an ABM which includes a high-resolution model of the environment and complex autonomous agents capable of perceiving the high-resolution environment [1, 2]. In order to meet the high computational demand of complex agents in high-resolution environment, we implemented a High-Performance Computing (HPC) extension. The HPC extension is scalable to efficiently simulate tens of millions of agents in several hundreds of square kilometers using thousands of CPU cores [3, 4, 5]. In this paper, we mainly focus on the modeling of car-car interactions and car-pedestrian interaction on narrow roads and at non-signalized junctions. Car trajectories at junctions are accurately modeled using B-splines, and speed profiles of cars at junctions are approximated match with the observations. The potential collision points of cars at junctions are accurately calculated, and car agents are programmed to avoid collisions with cars and pedestrians. As a demonstrative application, a hypothetical tsunami triggered evacuation scenario is simulated assuming the traffic signals are rendered non-functioning by the preceding earthquake.

The rest of this report is organized as follows. The section 2 presents short description of the developed ABM, and the agents' movements and actions at non-signalized junctions. Section 3 presents a short introduction to the HPC enhancements of the developed ABM, and the section 4 presents the demonstrative application of tsunami evacuation with non-functioning traffic signals.

2. Agent based system

To facilitate modeling of uncommon scenarios as an emergent behavior, the developed ABM composed of a 1m×1m resolution model of environment (see Fig. 1) and agents are programmed to perceive the features of the grid like roads, obstacles, debris, water bodies, etc. The grid environment is updated at suitable time intervals to mimic physical changes in the real environment like inundation, debris, etc. Each agent scans the grid within its sight distance to perceive the features, including the physical changes. The connectivity of walkable spaces in the grid environment is modeled as a network (see Fig. 1), and each agent stores their experiences, like the encountered blocked roads, congestions, etc., in its memory with reference to this network. Agents use their past experiences in making decisions like finding fastest route to an evacuation area. Since the network serves as the base map in each agent's memory, unlike the grid, the network is not updated according to the physical changes in the environment. The network is equipped with several functions to find paths with various requirements (e.g., minimize the use of narrow roads in nighttime evacuation scenarios).

Agents are assigned heterogeneous properties, like speed, sight distance, access to information, etc., to model heterogeneous crowds. Different types of agents are composed of different collections of constituent functions to mimic evacuees on foot and cars, volunteers, etc. The constituent function for collision avoidance is implemented adopting the Optimal Reciprocal Collision Avoidance (ORCA) [6] with some modifications and additional parameters to model interactions among agents [1, 2]. The parameters of are tuned to model avoid of collision among pedestrians and pedestrians, cars and cars, cars and pedestrians, and reproduce the relevant fundamental diagrams (i.e. speed vs. density graphs) as emergent phenomena. This is one of key point of the developed ABM. Unlike widely used 1D flow network models, where the agents' speeds are adjusted according to the average density in each link and suitable fundamental diagrams, the agents of the developed ABM are tuned such that characteristics like fundamental diagrams are produced as an emergent behavior from individual agent's actions [1, 2]. This allows us to model uncommon scenarios as an emergent behavior, with a reasonable confidence.

2.1 Un-signalized junction

During mass evacuations triggered by natural disasters like tsunami, it is unlikely that common everyday road conditions will prevail. Instead, it is reasonable to assume widespread presence of many uncommon conditions like narrow roads packed with people, evacuees walking on main roads outside the pedestrian paths and crossing the junction without respecting traffic signals, scattered debris, broken traffic lights due to the earthquake, etc. These scenarios are difficult to be simulated with existing simplified models, and ABMs with necessary agent functionalities to produce these as emergent behaviors are ideal for these scenarios. While our collision avoid algorithm, mentioned in the previous section, allows us to model these uncommon conditions occurring on straight roads as an emergent behavior, a few more elements are required to model these conditions occurring at junctions, bends, etc. The rest of this section presents some details of the required additional components like approximating trajectories and speed profiles of cars at junctions, estimating the potential collision points of cars on curved trajectories, controlling speed to avoid collisions with cars or pedestrians at unsignalized junctions, etc. We assume that drivers and pedestrians behave rationally without panicking.

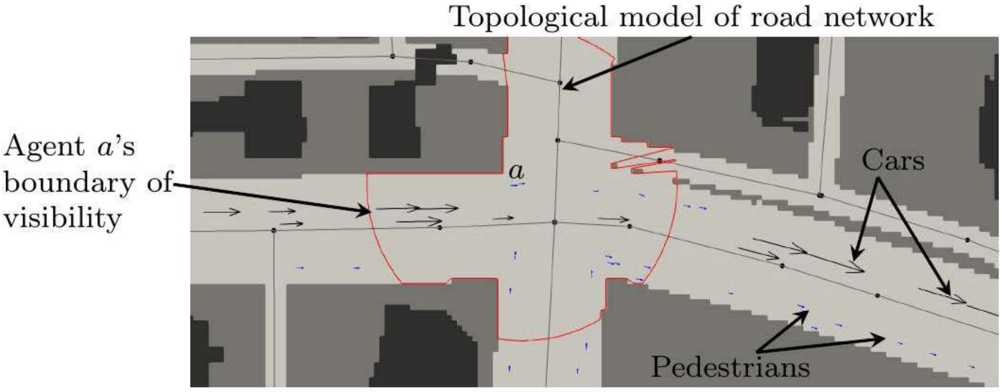


Fig. 1 An example of the grid and graph environment with snapshot of agents' movements at a junction. Blue and black arrows indicate instantaneous velocities of pedestrians and cars, respectively. Pedestrian agents walk along the edges, if the road can accommodate vehicles.

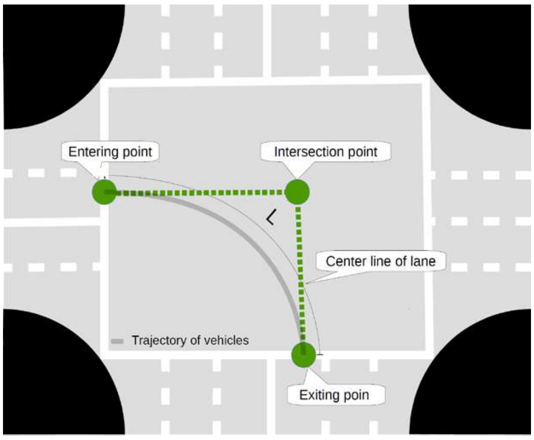


Fig. 2: Vehicle trajectories at intersections are approximated with third order B-spline curves

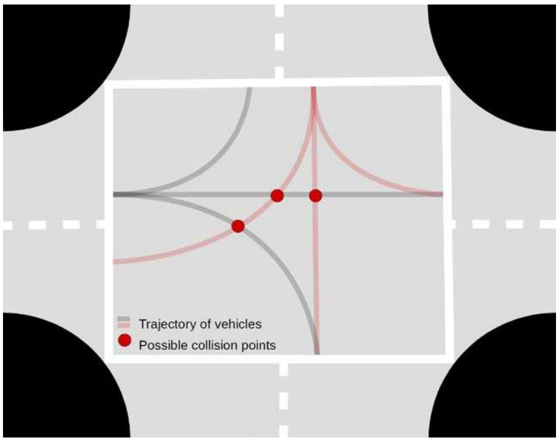


Fig. 3: Intersection points of multiple trajectories

2.1.1 Approximating vehicle trajectories at intersections

Our analysis of vehicle trajectory observations by Alhajyaseen et. al.[7] shows that vehicle trajectories can be easily approximated with B-splines. Vehicle trajectories at most common intersection geometries, except U-turns, can be approximated using B-spline with knot vector $[0,0,0,1,1,1]$ and three easy to find control points; 1) the lane center at the entry to intersection, 2) the intersection point between center lines of the incoming and outgoing lane, 3) lane center at the exiting point (see Fig. 2). Comparisons with the approximation proposed by Alhajyaseen et. al.[7] showed that our B-spline approximation deviates only by a few centimeters, which is negligible for this particular application. Though we found that a more accurate approximation can be made using NURBS, we use B-spline since the points along vehicle trajectories can be efficiently calculated with B-splines compared to NURBS; the above B-spline knot vector defines a simple third-order Bezier basis functions.

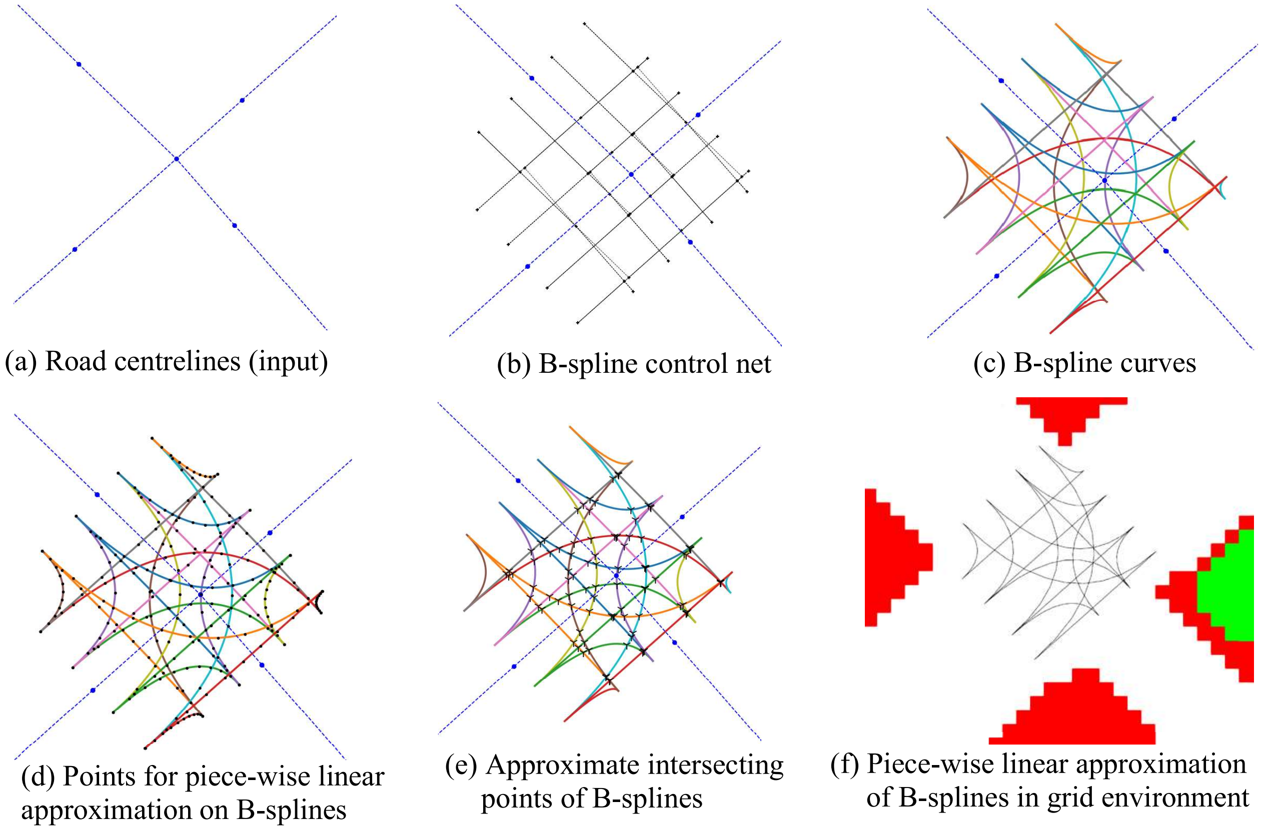

(a) Road centrelines (input)
(b) B-spline control net
(c) B-spline curves
(d) Points for piece-wise linear approximation on B-splines
(e) Approximate intersecting points of B-splines
(f) Piece-wise linear approximation of B-splines in grid environment

Fig. 4: Main steps involved in approximating car trajectories at a junctions

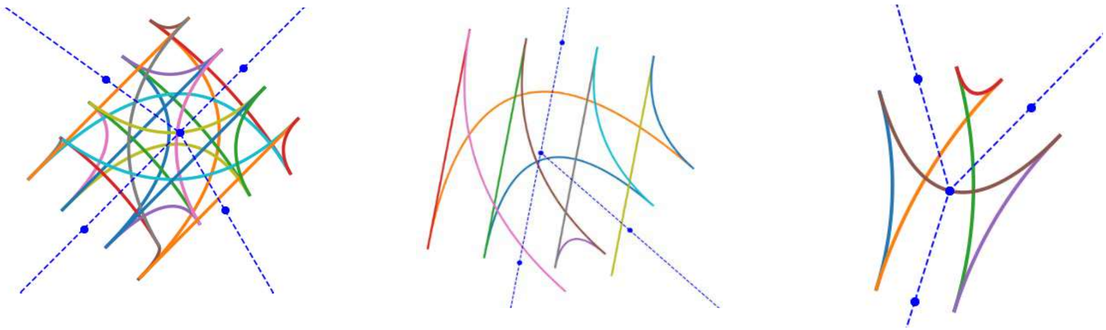


Fig. 5: B-spline approximations for junction of different configurations.

To further reduce the amount of computations in making car agents move along B-spline trajectories, we made a piece-wise linear approximation of B-splines curves using a few N number of points on the B-spline curves and assuming the trajectory between consecutive points to be linear (see Fig. 4d). The potential collision points of cars on different trajectories are defined based on the intersection points of B-spline trajectories (see Fig. 3). Since computing exact intersection point B-spline curves is complicated, we use the intersection points of the above piece-wise linear approximations of B-splines. We found that N=12 provides sufficiently accurate results. Figure 4 illustrates the main steps involved in modeling the car trajectories at junctions. Figure 5 shows our B-spline approximations for junctions of different configurations.

2.3 Speed profile of a vehicle at intersections

Not only accurate trajectories, but also realistic free flow speed profiles along these curved trajectories are essential to accurately model car-car interaction and car-pedestrian interactions. Since car's speed are usually one order of magnitude larger than that of pedestrians, unrealistic or sudden changes of car speeds can make pedestrian agents to behave abnormal. According to Dias et al. [8], speed profiles of vehicles can be approximated with higher order polynomial curves. While their approximation requires field observations, we use the following third order polynomial approximation which can be defined with three known parameters and one constraint.

$$v(x) = (-4V_a + 4V_d)\left(\frac{x}{L}\right)^3 + (8V_a - 4V_{min} - 4V_d)\left(\frac{x}{L}\right)^2 + (-5V_a + 4V_{min} + V_d)\left(\frac{x}{L}\right) + V_a, \qquad (1)$$

where $x \in [0, L]$ is the length measured along the trajectory, and L is the length of the trajectory. We assumed that at the point of the highest curvature, which we assumed to be at $x = \frac{L}{2}$ in the above case, the acceleration is zero (i.e., $dv/dx = 0$) and the car reaches the minimum speed of V_{min}. V_{min} can be considered as the maximum allowable speed to prevent accidents due to centripetal force, which can be determined based on the mass of a car and curvature of the trajectory. $V_a = v(0)$ is the car's approaching speed to the junction, and $V_d = v(L)$ is the desired departing speed. One can easily derive the equations if the maximum curvature is not at $x = \frac{L}{2}$. We found this approximation is can closely reproduce the observed speed profiles of cars at junctions. Equation (1) defines only the free flow speed of a given car agent along curved trajectories. If a car agent detects potential collision with another car or pedestrians, it decelerates to a speed $u_i(x) < v(x)$ so that it can prevent collisions, and accelerates to the free flow speed defined by Eq. (1) once it is clear of any collisions.

2.4 Car-car interactions at un-signalized at interactions

We classify car-car interaction at a junction to three groups; intersecting trajectories (e.g. V1 and V3 in Fig. 6), diverging trajectories, and merging trajectories (e.g. V2 and V4 in Fig. 5). In this model for simulating interactions at unsignalized intersections, we assume the drivers do not panic, and restrict themselves to the standard trajectories at junctions. We further assume

1. when approaching an intersection, cars reduce the speed to a comfortable speed of $V_a = v(0)$
2. each car observes neighbor cars' positions, their turn signals, and estimates their relative speeds
3. if a car identifies a potential collision, it avoids collision by applying comfortable deceleration to maintain a safe distance in between.

The implemented algorithm to resolve collision at junctions is somewhat complicated. However, it is based on the simple rule that the car to first arrive at the point of collision is given priority to move uninterrupted by the others. Given the length l_c and width w_c of a car, the locations at which two cars on the same or different

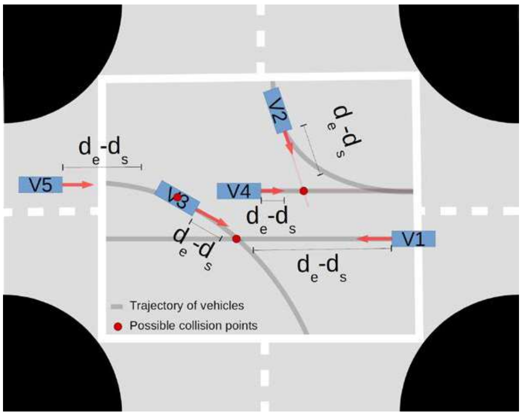
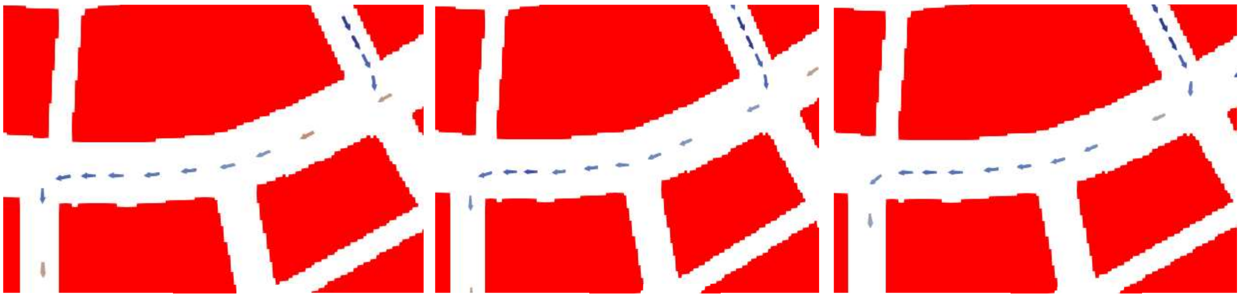


Fig. 6: Illustration of the three different types of collisions considered: V1-V3 are on intersecting trajectories; V2-V4 are on merging trajectories; V5-V3 are on same trajectory.

trajectories touch each other are defined as the points of collision. In case of merging trajectories, once the cars V2 and V4 (see Fig. 5) enter the merged road, they drive maintaining a safe gap between them. The safe gap between two vehicles on same lane is calculated considering time to decelerate including driver's reaction time, and an additional safe distance which is set to be the length of a car l_c. Figure 6 shows some snapshots of many cars at a junction. Since we assume that traffic lights are not functioning, cars entering junctions from different directions reduce their speeds according to the priorities we set, and that leads to traffic jams.

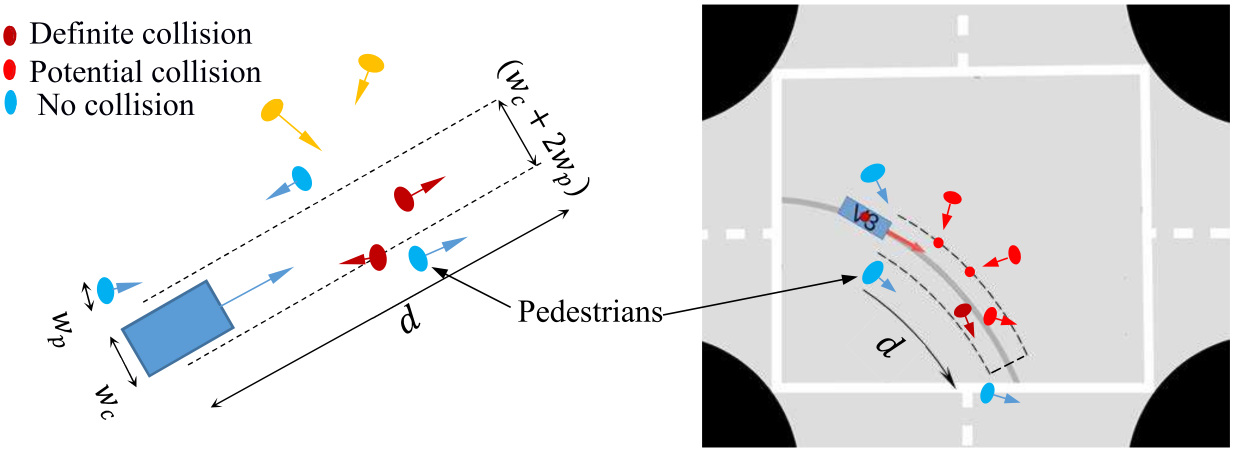
2.5 Car-pedestrian interaction

In case of car-pedestrian interaction, cars are set to give priority to pedestrians. First, a car agent estimates the distance d, which it must travel before stopping at a comfortable deceleration (see Fig. 7). Then, it identifies the pedestrian agents, shown in red color, ahead on its projected path of length d. Also, the car agent calculate which pedestrian agents (shown in colour green) can enter its projected path of length d, and identify which of these pedestrian agents can collide with the car according to their relative speeds. Then the car decelerates to avoid these potential collisions. Although the basic logic involved are the same, the calculations for curved car trajectories at junctions is somewhat complicated. Figure 8 shows some snapshots of cars slowed down at junctions by the passing pedestrians.



(a) (b) (c)

Fig. 6: Snapshots of the simulated car-car interaction at a junction.



(a) (b)

Fig. 7: A car first calculate the required distance, d, for it to decelerate to stop, calculates which pedestrians will be in its predicted path of length d, and decelerate to prevent collision with them. The agents shown in red colour are ahead on the path of car, while the agents shown in green are can enter the projected path of length d. w_c and w_p are width of a car and safe gap.

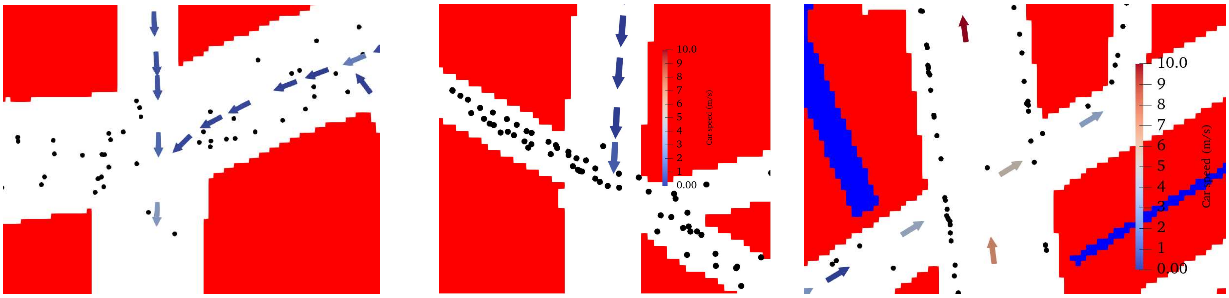
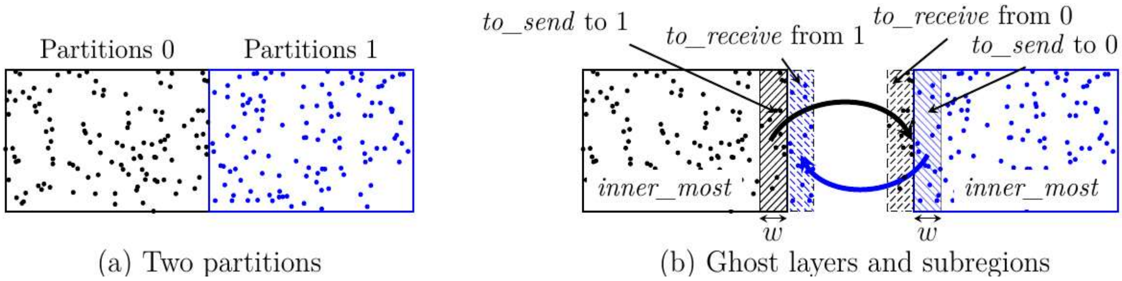


Fig. 8: Simulated car-pedestrians interaction at a junction



(a) Two partitions (b) Ghost layers and subregions

Fig. 9: Illustration of kd-tree based domain decomposition ,and maintaing continutity using ghost updates

3. HPC extension to meet the high computational demand

To meet the high computational demand of the autonomous agents, a scalable High-Performance Computing (HPC) extension was developed. Most of the basic constituent functions of agents, like *identify_grid_features, identify_debris_and_obstacles*, *car_pedestrian_intaraction_at_junctions*, etc. are computationally expensive. As an example, *identify_grid_features* of an agent scans its grid neighborhood within its sight distance in high resolution using a computation and memory intensive ray tracing algorithm. *identify_debris_and_obstacles* is used to identify fallen debris, etc. which are partially or completely blocking roads. The computation and memory intensive natures of the agents' constituent functions demand significant computational resources. As an example, 50 minutes long evacuation simulation of 90,000 agents with the environment shown in Fig. 9 at 0.2s time steps required 33 node-hours in K-computer; a computing node of K-computer consists of an 8-core SPARC64 VIIIfx processor with 16GB of RAM. To meet this high computational demand, a hybrid parallel extension was developed [1, 3, 4]. This section presents some details of the implemented HPC extension.

To assign nearly equal workloads to each CPU, the agents are partitioned into N number of geographically continuous rectangular subsets using kd-tree. The execution time for each agents depends on its type, visible surrounding, agent density it its neighborhood, etc. Hence, we use the measured execution time of each agent in generating kd-tree to assign nearly equal workload to each partition. Each partition is assigned to an MPI process, which is a shared memory compute node in our case, and the computations within a node are accelerated using OpenMP threads. Most of the algorithms of our interacting agents have extensive race conditions making it difficult to exploit shared memory parallelism. We use a data redundant approach to avoid most of these race conditions and OpenMP's *task* level parallelism to accelerate computations using shared memory parallelism. Figure 9 shows the domain decomposed for two MPI processes. Since an OpenMP thread in one compute node cannot access the memory of the compute nodes holding neighboring partitions, we maintain overlapped boundary regions, which are usually called ghost regions (see Fig. 9). The states of the agents lying in ghost regions are exchanged using MPI at each time step to make the agents in different partitions interact, and thereby maintaining the continuity of the simulation over multiple compute nodes. Since agents' actions are affected by the information within their visible surrounding, the thickness, $\boldsymbol{w}$, of the ghost region is set to be larger than the longest eyesight of the agents.

3.1 Communication hiding

The inter-node MPI communication required for maintaining the continuity is an extra overhead, and this overhead increases with the number of partitions or MPI processes. Communication hiding, which is doing some useful computation while communications are happening in the background, is a standard technique to minimize the performance degradation due to this communication overhead. To hide communications, we group the agents in an MPI process into three mutually exclusive sets. $\boldsymbol{to\_send}$ includes all the agents to be sent to neighbor MPI processes, $\boldsymbol{to\_recv}$ group includes all the agents in ghost regions, while the rest are included in $\boldsymbol{inner\_most}$ (see Fig. 9). We hide the communication overhead by first executing the agents in $\boldsymbol{to\_send}$, and posting non-blocking MPI calls to pass their updated states to the corresponding $\boldsymbol{to\_recv}$ regions of the neighboring MPI processes. While the communications progress in the background, the agents in $\boldsymbol{inner\_most}$ region are executed. If execution of $\boldsymbol{inner\_most}$ set takes longer than the time to complete the communication, we can effectively eliminate the communication overhead.

3.2 Dynamic load balancing

The movements of agents make them move from one partition to another, which we refer as $\boldsymbol{agent\ migration}$ in this paper. Permanently moving an agent from a partition to another is time consuming. At the same time, the workload assigned to a MPI process increases or decreases depending on the number of agents migrated in or out. If the differences in workloads are high, the parallel computing efficiency can significantly degenerate. When the load imbalance among MPI processes reaches a critical state, domain is re-partitioned to re-assign equal workloads to MPI processes. We measure the total execution time of the agents in each MPI process and repartitioning is called when the time for repartitioning process is smaller than the computational time wasted due to load imbalance.

Strategies like data redundant approach for eliminating data races, communication hiding, combining communication of multiple data, etc. enabled us to implement a scalable MPI-OpenMP hybrid parallel computing code with a high strong parallel scalability. Our simulation of a hypothetical evacuation event involving 10 million agents in 588km^2 area of central Tokyo in 1m×1m resolution produced 82% strong parallel scalability with 2048 nodes (2048×8 CPU cores) of the K supercomputer. According to the authors' knowledge, such large-scale simulations for this class of problems have not been reported in literature. Our code is designed such that the same code can be used in a modern note-PC to simulate a few ten thousand of agents efficiently.

4. Demonstrative simulations

As a demonstrative application of the presented agents' interactions at non-signalized junctions, we simulated hypothetical evacuation scenarios using Kochi city environment (see Fig. 10). We assume that people must evacuate to above 10m elevation; this hypothetical setting is around 30 times larger than the tsunami height mentioned in Kochi city office's homepage. It is assumed that emergency evacuation is advised at 11 p.m., and a total of 61,218 people are expected to evacuate to above 10 elevation. The locations of people at 11 p.m. are set according to the locations and area of buildings (see Fig. 10b). To quantitatively evaluate the effect of allowing cars for evacuation, we made a given p percentage of people to use cars assuming each car carries 3 persons. The slowest people living at least 1 km away from the nearest evacuation are converted to cars. We consider the following 4 scenarios.

(1) All the people walk
(2) 6% of the slowest are allowed to use cars; total of 1224 cars
(3) 9% of the slowest are allowed to use cars; total of 1836 cars
(4) 15% of the slowest are allowed to use cars; total of 3060 cars

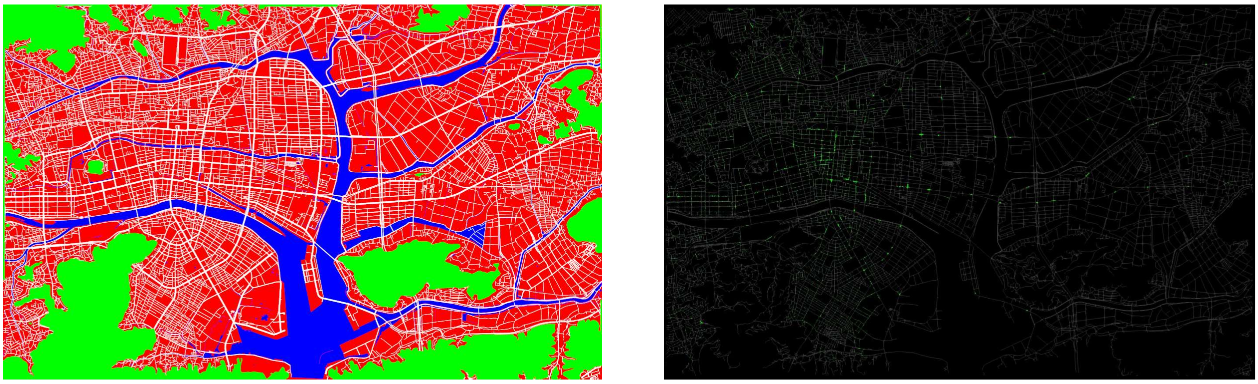

(a) Grid environment (b) Locations of people at 11 p.m.

Fig. 10: Kochi city environment used for the simulation. The areas shown in green, which are above 10m elevation, are considered safe evacuation areas.

The statistics of the speeds and evacuation start times assigned to pedestrian and car agents are shown in Fig. 11. Since this is a nighttime evacuation, we assumed that people preferred to walk on roads wider than 4m; the pedestrian agents are allowed to use roads of any widths, but they are programmed to give higher preference to use wider than 4m roads when available. Fig. 12 shows the expected number of pedestrians and cars for the scenario 3. Further, we assume that the earthquake, which generated the tsunami, has rendered the traffic signals non-functioning. The model we presented in section 2 is used to model the car-car and car-pedestrian interactions on roads and at the unsignalized junctions. We assume the car users behave rationally and follows decent driving habits. The car agents are assumed to give priority to pedestrian agents.

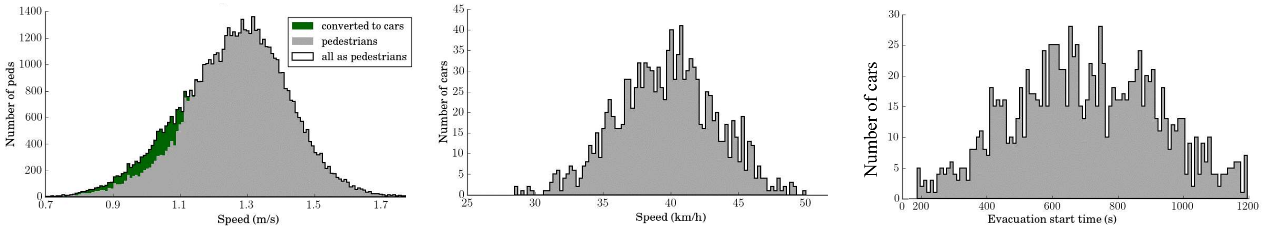
Figure 13 compares each scenario's progress of evacuation. According to Fig. 13c, when no cars are allowed, a small percentage of the population could not reach their evacuation area. This small percentage is the slow-moving people who were more than 1km away from the nearest safe area. On the other hand, allowing 6% of the population to use cars enables 100% of the population to reach their destinations. In addition, 6% use of cars have slightly accelerated the evacuation progress. While the use of 9% and 15% of cars accelerate the evacuation progress during the first 40 minutes, thereafter the large number of pedestrians on roads start to make the cars move slower (see Fig. 13b). We noticed that the pedestrians significantly influence the cars especially at junctions.

Figure 12a and 12b show the paths taken by the pedestrians and cars. As seen, the major routes used by cars are also shared by pedestrians, which can be a source of traffic jams. In fact, as seen in Fig. 12c the major traffic congestions occur on those major routes shared by cars and pedestrians. It is possible to further increase the number of cars users by introducing a policy to separate the major routes used by cars and pedestrians.

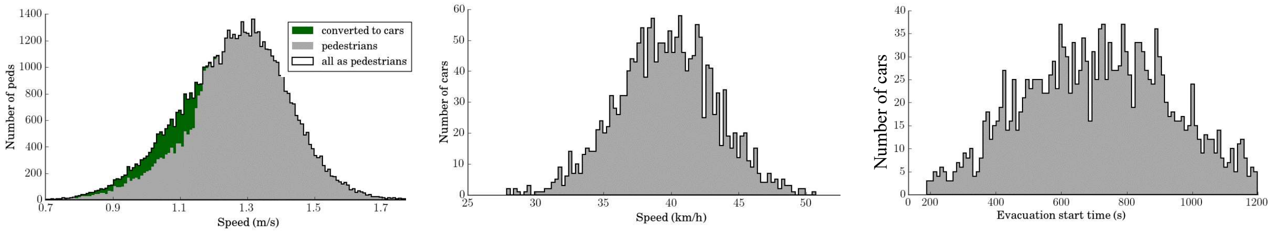
5. Discussion and concluding remarks

Agent Based Models with high resolution model of environment are versatile tools for studying the effect of uncommon conditions like broken traffic lights, low lighting conditions in night, debris scattered over roads, etc. during an emergency mass evacuation. While developing the logic to make the agents mimic the desired human behaviors is challenging, the resulting algorithms are both computationally demanding and memory intensive. It is necessary to efficiently utilize high performance computing infrastructures to simulate large urban areas using such ABMs. In this paper, we briefly presented the necessary components to simulate car-car and car-pedestrian interactions at unsignalized junctions, and the some of the strategies to develop scalable HPC extensions for simulating large urban areas like entire Tokyo. Simulating a hypothetical evacuation event involving 10 million agents in 588km^2 area of central Tokyo we demonstrated that the developed simulator scales at least up to 2048 compute nodes (2048×8 CPU cores) of the K supercomputer. The code is designed such that it can be used on a range of hardware starting from an ordinary note-PC to a supercomputer, depending on the size of the problem. Simulating hypothetical evacuation scenario involving mixed car-pedestrian mode evacuation, we demonstrate that the use of certain percentage of cars can be beneficial, provided that mainly the needy are allowed to use cars and the time window for the car usage are restricted. Further simulations with strategies to reduce traffic jams induced by large number of pedestrians on roads will enable one to find more efficient means of evacuating. Development of such ABMs with high resolution model of environment and sound logic to mimic evacuees' behaviors will enable to produce digital twins with which the effectiveness of emergency mass evacuation plans can be quantitatively estimated and efficient strategies to smoothly evacuate large urban areas can be discovered.

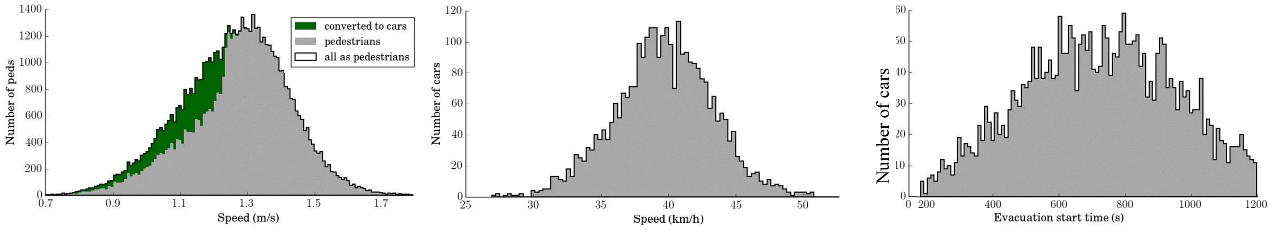
(a) Scenario 1 with 0% car users



(b) Scenario 2 with 6% car users

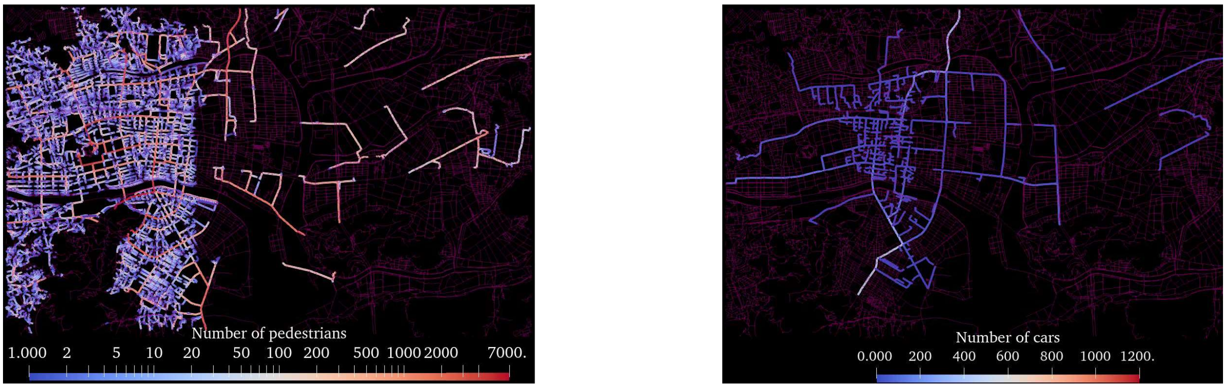


(c) Scenario 3 with 9% car users



(d) Scenario 4 with 15% car users

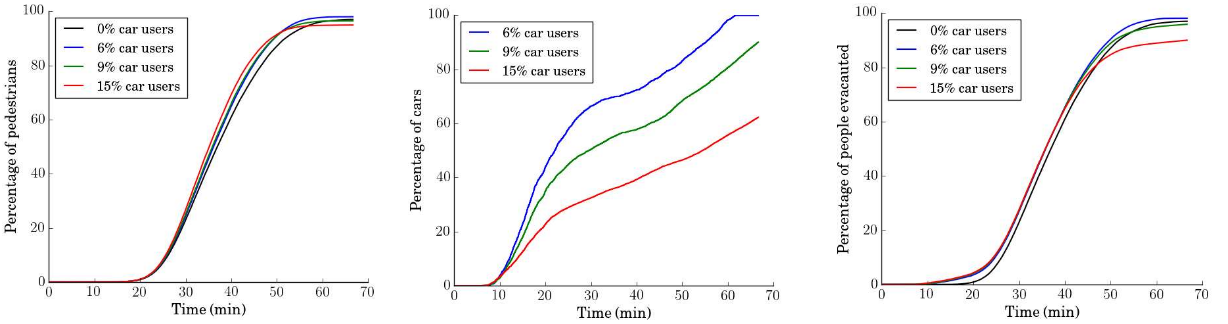
Fig. 11: Speed characteristic and evacuation start time of pedestrians and cars for each scenario. The portions shown dark green in left side graphs are converted to cars, putting three people in each car. The evacuation start time distribution of pedestrians for every scenario is very close that of the scenario 1.



(a) Roads taken by pedestrians (b) Roads taken by cars

(c) Roads with slowest car speed (traffic jams)

Fig. 12 Paths taken by pedestrians and cars in scenario 3. Colors indicate the number of agents traversed.



(a) (b) (c)

Fig. 13: Comparison of evacuation time histories of the 4 scenarios considered.

6. Acknowledgements

Authors would like to thank Professor Hideki Nakamura at Nagoya University for authorizing us to utilize the trajectory data for validating our B-spline approximation. We thank Tokyo Electric Power Services Co., Ltd. (TEPSCO) and Tokyo Electric Power Company (TEPCO) for their generous funding to support the development of the codes. Some of the results are obtained using K computer at the RIKEN Advanced Institute for Computational Science (hp150283).

7. References

[1] Leonel Aguilar, Maddegedara Lalith and Muneo Hori, 2017, Time critical mass evacuation simulation combining a multi agent system and high-performance computing, *Multi Agent Systems*, ISBN 978-953-51-5504-1, DOI: 10.5772/intechopen.69844

[2] Melgar LEA, Wijerathne L, Hori M, Ichimura T and Tanaka S, 2013. A Scalable Workbench for Large Urban Area Simulations, Comprised of Resources for Behavioural Models, Interactions and Dynamic Environments. *Lecture Notes in Computer Science*, **8861**, 166-181.

[3] Leonel Aguilar, Maddegedara Lalith, Tsyoshi Ichimura, and Muneo Hori, 2017, On the performance and scalability of an HPC enhanced MAS based evacuation simulator, *Procedia Computer Science*, **108**, 937-947.

[4] Lalith Wijerathne, Wasuwat Petprakob, Leonel Aguilar, Muneo Hori, Tsuyoshi Ichmura, 2018, Scalable HPC Enhanced Agent Based System for Simulating Mixed Mode Evacuation of Large Urban Areas, *Transportation Research Procedia*, **34**, 275-282, ISSN 2352-1465, https://doi.org/10.1016/j.trpro.2018.11.042.

[5] Wijerathne ML, Melgar LA, Hori M, Ichimura T, and Tanaka S, 2013, HPC enhanced large urban area evacuation simulations with vision based autonomously navigating multiagents, *Procedia Computer Science*, **18**, 1515-152

[6] Berg JVD, Guy SJ, Lin M, and Manocha D, 2011, Reciprocal n-body collision avoidance," Robotics Research: The 14th International Symposium ISRR, *Springer Tracts in Advanced Robotics*, Springer-Verlag, **70**, 3-19.

[7] Alhajyaseen WK, Asano M, Nakamura H, Tan DM: Stochastic approach for modeling the effects of intersection geometry on turning vehicle paths. *Transportation Research Part C: Emerging Technologies,* **32**, 179 -192 (2013).

[8] Dias C, Iryo-Asano, M Oguchi T, 2017, Predicting optimal trajectory of left-turning vehicle at signalized intersection. *Transportation Research Procedia,* **21**, 240 -250. https://doi.org/10.1016/j.trpro.2017.03.093