Replacing Method for Multi-Agent Crowd Simulation by Convolutional Neural Network

Yu Yamashita[1,2], Shunki Takami[1,2], Shusuke Shigenaka[1,2], Masaki Onishi[2], and Atsuyuki Morishima[1]

[1] University of Tsukuba, Japan
yu.yamashita.2021b@mlab.info
morishima-office@ml.cc.tsukuba.ac.jp
[2] National Institute of Advanced Industrial Science and Technology, Japan
{s-takami,shusuke-shigenaka}@aist.go.jp
onishi@ni.aist.go.jp

Abstract. Multi-agent crowd simulations are used to analyze crowd flows. However, there is a critical problem that the computational time increases with the number of agents because these simulations are based on the interaction of the agents. The approach of using deep neural networks is effective in some applications, such as fluid dynamics simulations. We propose a method of using convolutional neural networks to estimate simulation results from the conditions of each agent and the initial arrangement of a crowd. We evaluated our proposed method through evacuation simulation and demonstrated that our proposed method could obtain evacuation simulation results with high speed.

Keywords: Multi-Agent Crowd Simulation · Convolutional Neural Network · Evacuation simulation

1 Introduction

Multi-agent crowd simulations are used to analyze crowd flows. These simulations can allow each agent to have conditions, such as a walking speed and route, and compute crowd flows under specific situations. For example, according to previous works, simulations can be used to compute crowd flows during the evacuation from buildings [7, 12, 27, 31]. While the computations in these works are non-interactive, they should be performed with high speed if they are to be used more adaptively.

There is a critical problem that the computational time increases with the number of agents because these simulations are based on the interaction of the agents. Obtaining results with high speed via these simulations is difficult. Nonetheless, no study has addressed this problem, and therefore, a method capable of obtaining simulation results with high speed is required.

The approach of using deep neural networks (DNNs) for simulations has been proposed in some fields, such as fluid dynamics simulations [11, 26] and molecular dynamics simulations [4]. These works have demonstrated that this approach can obtain simulation results with high speed, indicating that it is effective for the problem, and therefore, we use DNNs in this study.

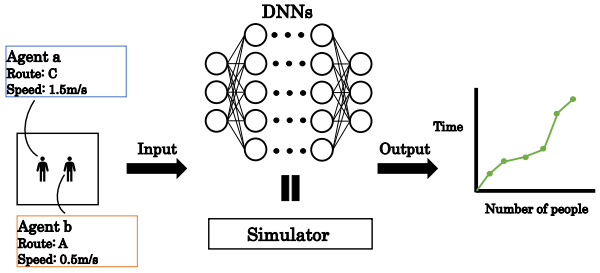

Fig. 1. Our problem setting

This work estimates simulation results from the conditions of each agent and the initial arrangement of a crowd. In this work, we assume the evacuation of a crowd from a building. Given the conditions of each agent (such as the route and the speed) and the initial arrangement of the crowd, we estimate the transition in the number of evacuees over time (Fig. 1). We propose a method that utilizes convolution neural networks (CNNs), which are DNNs. CNNs can extract effective spatial features of crowds and deal with multiple conditions (channels). In addition, we convert from the conditions of each agent and the initial arrangement of a crowd to a structure suited for CNNs. Our goal is to replace the simulation computation process with CNNs to obtain the results with high speed.

We evaluated the effectiveness of our proposed method in terms of estimation accuracy and speed. For this purpose, we used simulation data on evacuation settings in a theater. We show that our proposed method performs evacuation simulations 50 times faster than the simulator, with high accuracy simulation results. Our experimental results show that our proposed method can obtain simulation results with high speed.

Our contributions are summarized as follows:

1. We first address the critical problem of increasing computational time with the number of agents in multi-agent crowd simulations.
2. We propose a method that uses CNNs, which are DNNs, to tackle the above problem. CNNs can extract effective spatial features of crowds and deal with multiple conditions (channels).
3. We evaluated our proposed method through evacuation simulation and demonstrated that our proposed method could obtain evacuation simulation results with high speed.

2 Related Work

Several works have been conducted to compute traffic flows [2, 15, 20, 24], and human behaviors [16] through multi-agent simulations. In this work, we deal with multi-agent simulations to compute crowd flows (multi-agent crowd simulations) [29]. For example,

these simulations have computed the evacuation of crowd in buildings [7, 31], aircrafts [19], airports [27], and cities [12]. Simulation systems cosidering geographical information [28] and human behavioral parameters [18] are proposed. These simulations employ the bottom-up approach, which is computing crowd flows based on the interaction on the agents, and the computational time increases with the number of agents. Thus, it is not easy to use them if we want to obtain their results as soon as possible. Nonetheless, no works have addressed this problem, and therefore, we address this problem in this study.

Simulations are often time-consuming. The approach of using DNNs has been proposed in some applications to solve this problem. In fluid dynamics simulations, methods to replace part of the computational process of Navier-Stokes equations with CNNs are proposed [11, 26]. In molecular dynamics simulations, a method to estimate long-term simulation results from using short-term simulation results by using DNNs is proposed. These works have shown that this approach can obtain simulation results with high speed, which indicates it is effective for the problem. These simulations and the multi-agent crowd simulation are the same in that they compute complex phenomena composed of numerous elements, and therefore, we use DNNs in this study.

CNNs have made great progress mainly in the image domain (image recognition [23], image segmentation [1], object detection [17], and pose estimation [3]). These networks have convolutional and pooling layers, extract effective spatial features, and deal with multiple channels. They have recently been applied not only in the image domain but also in other fields, such as natural language processing [6], acoustic signal processing [14], and game AI [22]. Several works succeeded in predicting future crowd flows with high accuracy by extracting spatial features of the current crowds using CNNs [10, 13, 25, 30]. The initial arrangement of a crowd significantly impacts future crowd flows in multi-agent crowd simulations, and therefore, we use CNNs.

3 Problem Setting

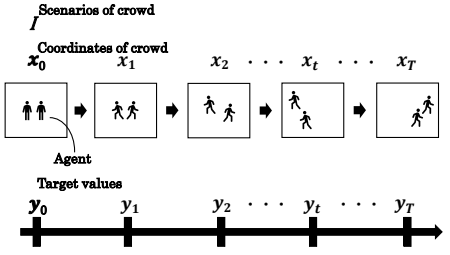

Fig. 2. General setting in multi-agent crowd simulation

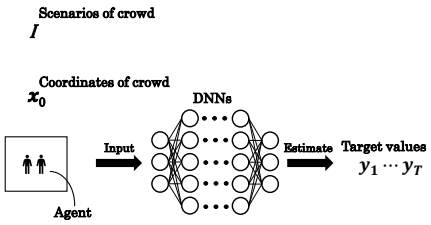

Fig. 3. Our setting

Figure 2 shows the general setting in multi-agent crowd simulation. The scenarios of the crowd $\boldsymbol{I}$ define the behavior of the crowd and are composed of multiple conditions

such as route and speed [5, 9]. The coordinates of the crowd $\boldsymbol{x}_{t \in T}$ indicates the position of the crowd at the timestamp $t \in T$ and has horizontal and vertical values, respectively. Given the scenarios of the crowd $\boldsymbol{I}$ and the initial coordinates of the crowd $\boldsymbol{x}_0$, the simulator computes the coordinates of the crowd $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T$ and outputs the target values $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$. The rectangle in Figure 2 shows the snapshot of the crowd created in the simulator based on the coordinates of the crowd $\boldsymbol{x}_t$.

4 Proposed Method

In this work, we estimate target values $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$ from the scenarios of the crowd $\boldsymbol{I}$ and the initial coordinates of the crowd $\boldsymbol{x}_0$ (Fig. 3). We propose a method of using CNNs to extract effective spatial features of crowds since the initial arrangement of a crowd has a significant impact on future crowd flows in multi-agent crowd simulations. In addition, we convert the conditions of each agent and the initial arrangement of a crowd to a three-dimensional tensor (3d-tensor) suited for CNNs. Our method consists of two steps: (1) converting the set of scenarios $\boldsymbol{I}$ and the initial coordinates $\boldsymbol{x}_0$ to a 3d-tensor and (2) estimating target values $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$ by using CNNs.

We use a multi-agent crowd simulator to generate several sets of pairs of scenarios $\boldsymbol{I}$, a set of initial coordinates $\boldsymbol{x}_0$, and target values $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$. These are used as a dataset for training CNNs.

4.1 Conversion to 3d-tensor

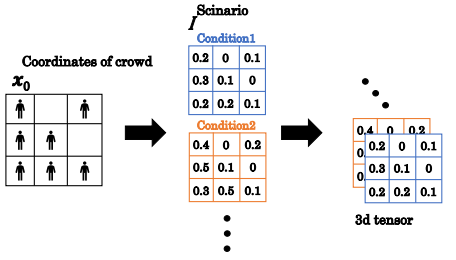

Fig. 4. Conversion to a 3d-tensor

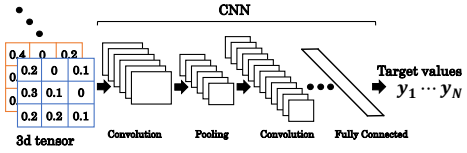

Fig. 5. Estimation by using convolutional neural networks (CNNs)

We convert the scenarios of the crowd $\boldsymbol{I}$ and the initial coordinates of the crowd $\boldsymbol{x}_0$ to a 3d-tensor (Fig. 4). First, we develop a two-dimensional tensor (2d-tensor) for each condition and each value, which is assigned an arbitrary value according to the conditions of the agent existing at that coordinate. Then, we develop 2d-tensors for the number of conditions and stack them in the channel direction to develop a 3d-tensor $\boldsymbol{z} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$. For example, if the scenario consists of speed, route, and departure time, we develop a 3d-tensor with three channels. A 3d-tensor is a structure suited for

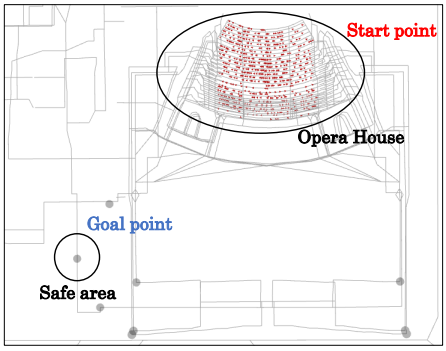

Fig. 6. Starting point and goal point of evacuation

CNNs and can retain features of the initial coordinates and the scenario composed of multiple conditions. These features are essential for estimating target values because simulations are based on them.

4.2 Estimation by using CNNs

CNNs output target values $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$ from the scenarios of the crowd $\boldsymbol{I}$ and the initial coordinates of the crowd $\boldsymbol{x}_0$ (Fig. 5). CNNs are composed of convolutional and pooling layers and are applied in several domains. CNNs have two advantages: (1) extracting effective spatial features of crowds and (2) dealing with multiple conditions (channels).

5 Experiment

5.1 Evacuation simulation

In this work, we assumed the evacuation of a crowd from a building. In the event of a disaster, the building manager wants to evacuate the crowd in the building to a safe area smoothly. The manager can design various evacuation guidance guidelines through simulations (such as estimating the elapsed time until the crowd has evacuated from the building); however, simulation processes are time-consuming. Our proposed method enables the manager to plan the best guidance for daily situations by reducing the computation time of simulations.

We assumed The New National Theatre, Tokyo, as the building in this study. We used CrowdWalk as a multi-agent crowd simulator. The evacuation of the crowd was computed based on an actual evacuation drill in the theater by using CrowdWalk [25]. The crowd is in the Opera House, which is defined as the start point, and evacuates to a safe area, which is defined as the goal point (Fig. 6).

The scenario is composed of four conditions: s_1 the time to start the evacuation, s_2 the exit door of the Opera House, s_3 the route for the evacuation, and s_4 the walking speed. s_1 to s_3 are based on the guidance by the staff of the theater. s_4 are dependent on the people characters, such as the elderly, children, the public, and wheelchair users.

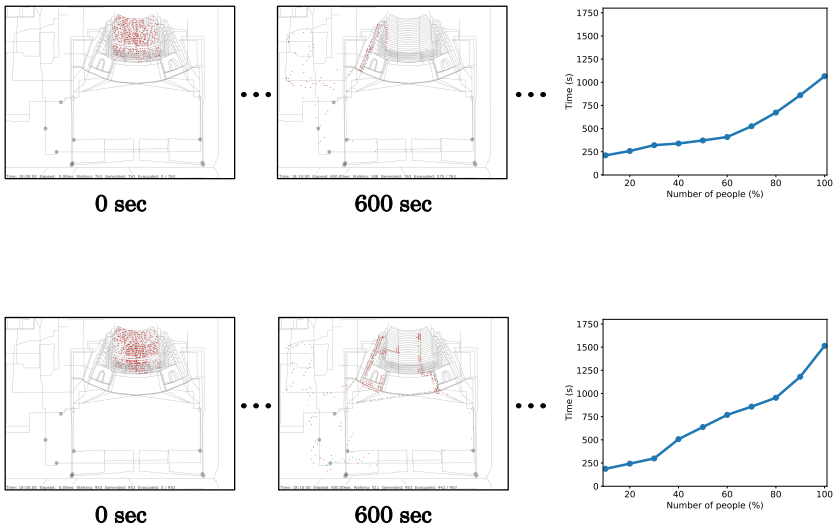

Fig. 7. Snapshots of an evacuation simulation and transitions in the number of evacuees over time

The condition s_1 indicates the time to start evacuation in seconds from a point in time after a disaster occurs. The target values $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$ indicate the elapsed times for the number of people who have arrived at the goal point to reach $10t\%$ of the total number of people. Let $T = 10$ and $0 \leq t \leq 10$. We indicate the transition in the number of evacuees over time by connecting the elapsed times $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$. Figure 7 shows snapshots of the crowd created in the simulator based on the coordinates of the crowd and transitions in the number of evacuees. Given the scenarios of the crowd $\boldsymbol{I}$ composed of four conditions and the initial coordinates of the crowd $\boldsymbol{x}_0$, the simulator computes the coordinates of the crowd $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T$ based on the scenarios of the crowd $\boldsymbol{I}$ and outputs the elapsed times (the transition in the number of evacuees over time) $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$. Therefore, we estimated the elapsed times $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$ from the scenarios of the crowd $\boldsymbol{I}$ and the initial coordinates of the crowd $\boldsymbol{x}_0$ in this experiment.

5.2 Dataset setting

We gave the various scenarios of the crowd and initial coordinates of the crowd and let the simulator output the elapsed times for this experiment. We obtained 60,000 sets of scenarios of the crowd, the initial coordinates of the crowd, and the elapsed times.

The Opera House has 22 horizontal columns and up to 42 vertical columns. Figure 8 and Figure 9 show the Opera House, divided by seats and blocks.

Initial coordinates of the crowd $\boldsymbol{I}$: For 30% of samples of all data, agents were generated randomly for each seat ranging from 0 to no greater than the maximum number of people allowed to sit in that seat (Fig. 8). For 70% of samples of all data,

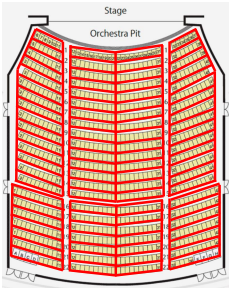

Fig. 8. Opera House divided by seats

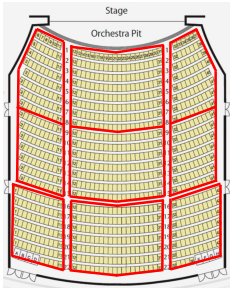

Fig. 9. Opera House divided by blocks

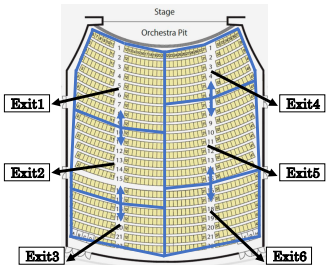

Fig. 10. Exit from Opera House

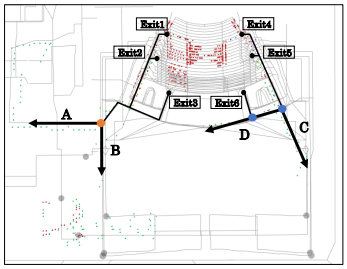

Fig. 11. Evacuation routes guidance

$\alpha \in \{0.4, 0.7, 1.0\}$ was chosen randomly for each block, and agents were generated randomly for seats in that block with the maximum number of people$\times\alpha$ (Fig. 9).

Time to start evacuation s_1: The time to start is within $\{0, 30, \cdots, 300\}$. For 10% of samples of all data, the time was set 0 s for all agents. For 20% of samples of all data, the time was chosen randomly for each seat (Fig. 8). For 70% of samples of all data, the time was chosen randomly for each block (Fig. 9). It was assumed that at least one seat agents start evacuation at 0 s.

Door to exit the Opera House through s_2: Six exiting doors are within Exit 1-6 (Fig. 10). Agents on the left half of the Opera House exited through one of Exit 1-3, and agents on the right half exited through one of Exit 4-6. Two boundaries were set randomly on the left half and right half, and agents sitting in front of the first boundary exited from Exit 1 (left half) or Exit 4 (right half), agents sitting between the first and second boundaries exited from Exit 2 (left half) or Exit 5 (right half), and agents sitting behind the second boundary exited from Exit 3 (left half) or Exit 6 (right half).

Route to evacuate through s_3: Agents exiting from the Opera House through Exit 1-3 evacuated through route A or B, and agents exiting through Exit 4-6 evacuated through

route C or D. The rates of routes A and B, along with C and D, were randomly set, and the route was chosen based on each rate.

Walking speed s_4: There are three-speed types: the default speed of CrowdWalk, 1.5 and 0.5 times faster than the default speed. We assumed that the first speed is the average speed of typical adults, the second speed is the average speed of young people, and the third speed is the average speed of older people or wheelchair users. For 30% of samples of all data, the speed was randomly chosen for each seat (Fig. 8). For 70% of samples of all data, the speed was randomly chosen for each block (Fig. 9).

5.3 Conversion to 3d-tensor

We converted the scenarios of the crowd $\boldsymbol{I}$ composed of four conditions and the initial coordinates of the crowd $\boldsymbol{x}_0$ to a 3d-tensor. First, we developed a 2d-tensor for each condition and value, which was assigned an arbitrary value according to the condition of the agent that existed at that coordinate. In a tensor of the time to start evacuation s_1, agents starting evacuation at 0, 30, ... , 300 s later were assigned 1, 2, ... , 11, respectively. In a tensor of the door to exit the Opera House through s_2, agents exiting through the doors Exit 1, Exit 2, ... , Exit 6 were assigned 1, 2, ... , 6, respectively. In a tensor of the route to evacuate through s_3, agents evacuating through routes A, B, C, and D were assigned 1, 2, 3, and 4, respectively. In a tensor of the walking speed s_4, agents walking at 0.5 times faster than the default speed, the default speed, and 1.5 times faster than the default speed were assigned 1, 2, and 3, respectively. The values of tensors with no agents were assigned 0, and we multiplied all values by 0.1 following the standard techniques of DNNs [21]. Then, we stacked these developed 2d-tensors in the channel direction to develop a 3d-tensor $\boldsymbol{z} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, thus we obtained a 3d-tensor of $4 \times 22 \times 42$.

5.4 Estimation using CNNs

CNNs output the elapsed times (the transition in the number of evacuees over time) from the 3d-tensor. We used ResNet-50 [8], which is a typical CNN. ResNet has a residual module that includes a shortcut to pass $H(\boldsymbol{a}) = F(\boldsymbol{a}) + \boldsymbol{a}$ to the next layer. The residual module has a shortcut that allows the gradient to be passed directly to lower layers during backpropagation. This prevents the gradient from vanishing even in a network with very deep layers and enables efficient learning.

We used root mean squared error (RMSE) as the error function of the output layer of the CNN and Adam as the optimization function with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The number of the epoch was set to 200, and the learning rate was first set to 0.001 and multiplied by 0.5 for 60, 120, and 160 epochs. The batch size was set to 256.

We split all data into training data (40,000), validation data (10,000), and test data (10,000).

6 Result

We evaluated the estimation accuracy, the multiple conditions dealing ability, and the estimation speed. We show that our proposed method can estimate simulation results

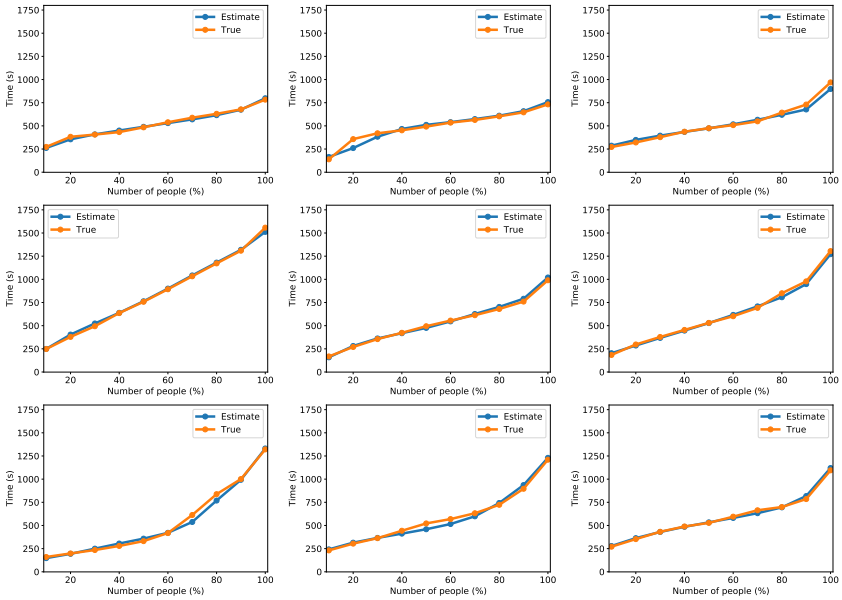

Fig. 12. Estimated and true elapsed times. The blue color indicates the estimated transition and the orange color indicates the true transition. The horizontal axis shows the percentage of people who have arrived at the goal point, and the vertical axis shows the elapsed times for each figure.

Table 1. Average loss for number of condition types

Number of condition types	Average loss (RMSE)
1	116.607
2	98.879
3	75.207
4	**34.157**

with high speed and accuracy and deal with multiple conditions based on experimental results.

First, we evaluated the estimation accuracy. The error (RMSE) for the test data was 34.157. Figure 12 shows six randomly selected estimated and true elapsed times (the transitions in the number of evacuees over time) for the test data. The blue and orange colors indicate the estimated true transitions, respectively. The horizontal axis shows the percentage of people who have arrived at the goal point, and the vertical axis shows the elapsed times for each figure. These results show that the estimated and true elapsed times almost overlap, indicating that the proposed method can estimate simulation results with high accuracy.

Second, we evaluated the multiple conditions dealing ability. The ability indicates whether our proposed method could train from multiple conditions. We compared the average loss (RMSE) when the number of condition types was varied from 1 to 4 (Table

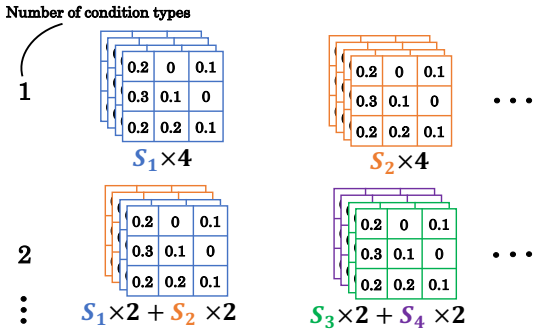

Fig. 13. 3d-tensors for number of condition types

Table 2. Time taken to output a result using the proposed method and the simulation

Simulator (CrowdWalk)	Our proposed method
8.223 (s)	**0.161 (s)**

1). The number of channels was set to be four even if the number of condition types was different (Fig. 13). For example, when the number of condition types is one, a 3d-tensor is composed of four s_1, s_2, or others. When the number of condition types is two, a 3d-tensor is composed of two s_1 and two s_2, two s_3 and two s_4, or others. We developed all possible combinations of conditions and averaged the loss (RMSE) for the number of condition types. The result shows that the average loss decreases as the number of conditions increases, and it is minimum for all conditions (the number of condition types is 4) These indicate that the proposed method can deal with scenarios composed of multiple conditions.

Finally, we evaluated the estimation speed. We compared the time taken to output a result using the proposed method and simulation (Table 2). The simulation was computed on an Intel i7-5930K CPU. From measurement results, it took approximately 8.223 s on average. Our proposed method was computed on a TITAN RTX with 24 GB. From measurement results, it took approximately 0.161 s on average. Therefore, the proposed method could estimate approximately 50 times faster than the simulation. This indicates that the proposed method could obtain the results with high speed. The estimation time of the proposed method is independent of the simulation time.

In this study, we use DNNs to estimate simulation results with high speed. DNNs perform with high accuracy in using large training data, target samples interpolated into the training data. Therefore, it is desirable to have large and comprehensive training data. A situation in which the training data are limited is a future challenge.

Based on the results, we conclude that our proposed method replaces the simulation computation process and enables us to rapidly obtain simulation results.

7 Conclusion

We addressed the critical problem that the time to compute increases with the number of agents in multi-agent crowd simulations. We proposed a method using CNNs to estimate simulation results from the conditions of each agent and the initial arrangement of the crowd. We evaluated our proposed method via evacuation simulations and demonstrated that our proposed method enables us to obtain evacuation simulation results with high speed.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
2. Balmer, M., Cetin, N., Nagel, K., Raney, B.: Towards truly agent-based traffic and mobility simulations. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004. pp. 60–67. IEEE (2004)
3. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7291–7299 (2017)
4. Endo, K., Tomobe, K., Yasuoka, K.: Multi-step time series generator for molecular dynamics. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
5. Feng, Y., Duives, D., Daamen, W., Hoogendoorn, S.: Data collection methods for studying pedestrian behaviour: A systematic review. Building and Environment **187**, 107329 (2021)
6. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: International Conference on Machine Learning. pp. 1243–1252. PMLR (2017)
7. Gianni, D., Loukas, G., Gelenbe, E., et al.: A simulation framework for the investigation of adaptive behaviours in largely populated building evacuation scenarios. In: Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 08), Estoril, Portugal. vol. 1216. Citeseer (2008)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Hoogendoorn, S.P., Bovy, P.H.: Pedestrian route-choice and activity scheduling theory and models. Transportation Research Part B: Methodological **38**(2), 169–190 (2004)
10. Jiang, R., Song, X., Huang, D., Song, X., Xia, T., Cai, Z., Wang, Z., Kim, K.S., Shibasaki, R.: Deepurbanevent: A system for predicting citywide crowd dynamics at big events. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2114–2122 (2019)
11. Kochkov, D., Smith, J.A., Alieva, A., Wang, Q., Brenner, M.P., Hoyer, S.: Machine learning–accelerated computational fluid dynamics. Proceedings of the National Academy of Sciences **118**(21) (2021)
12. Lämmel, G., Rieser, M., Nagel, K.: Bottlenecks and congestion in evacuation scenarios: A microscopic evacuation simulation for large-scale disasters. In: Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal (2008)
13. Liang, Y., Ouyang, K., Jing, L., Ruan, S., Liu, Y., Zhang, J., Rosenblum, D.S., Zheng, Y.: Urbanfm: Inferring fine-grained urban flows. In: proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & Data Mining. pp. 3132–3142 (2019)

14. Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G., Lockhart, E., Cobo, L., Stimberg, F., et al.: Parallel wavenet: Fast high-fidelity speech synthesis. In: International conference on machine learning. pp. 3918–3926. PMLR (2018)
15. Paruchuri, P., Pullalarevu, A.R., Karlapalem, K.: Multi agent simulation of unorganized traffic. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1. pp. 176–183 (2002)
16. Rasouli, A.: Pedestrian simulation: A review. arXiv preprint arXiv:2102.03289 (2021)
17. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
18. Sharma, S.: Avatarsim: A multi-agent system for emergency evacuation simulation. Journal of Computational Methods in Sciences and Engineering **9**(s1), S13–S22 (2009)
19. Sharma, S., Singh, H., Prakash, A.: Multi-agent modeling and simulation of human behavior in aircraft evacuations. IEEE Transactions on aerospace and electronic systems **44**(4), 1477–1488 (2008)
20. Sharon, G., Hanna, J.P., Rambha, T., Levin, M.W., Albert, M., Boyles, S.D., Stone, P.: Real-time adaptive tolling scheme for optimized social welfare in traffic networks. In: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2017) (2017)
21. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**(1), 1–48 (2019)
22. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. nature **550**(7676), 354–359 (2017)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Singh, A.J., Nguyen, D.T., Kumar, A., Lau, H.C.: Multiagent decision making for maritime traffic management. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 6171–6178 (2019)
25. Takeuchi, K., Nishida, R., Kashima, H., Onishi, M.: Grab the reins of crowds: Estimating the effects of crowd movement guidance using causal inference. arXiv preprint arXiv:2102.03980 (2021)
26. Tompson, J., Schlachter, K., Sprechmann, P., Perlin, K.: Accelerating eulerian fluid simulation with convolutional networks. In: International Conference on Machine Learning. pp. 3424–3433. PMLR (2017)
27. Tsai, J., Fridman, N., Bowring, E., Brown, M., Epstein, S., Kaminka, G.A., Marsella, S., Ogden, A., Rika, I., Sheel, A., et al.: Escapes: evacuation simulation with children, authorities, parents, emotions, and social comparison. In: AAMAS. vol. 11, pp. 457–464 (2011)
28. Uno, K., Kashiyama, K.: Development of simulation system for the disaster evacuation based on multi-agent model using gis. Tsinghua Science and Technology **13**(S1), 348–353 (2008)
29. Yamashita, T., Okada, T., Noda, I.: Implementation of simulation environment for exhaustive analysis of huge-scale pedestrian flow. SICE Journal of Control, Measurement, and System Integration **6**(2), 137–146 (2013)
30. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-first AAAI conference on artificial intelligence (2017)
31. Zhou, M., Dong, H., Ioannou, P.A., Zhao, Y., Wang, F.Y.: Guided crowd evacuation: approaches and challenges. IEEE/CAA Journal of Automatica Sinica **6**(5), 1081–1094 (2019)