Journal of AI and Data Mining
Vol 7, No 3, 2019, 403- 409
DOI: 10.22044/JADM.2018.6854.1808

A Data-driven Method for Crowd Simulation using a Holonification Model

J. Peymanfard and N. Mozayani*

School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

Received 13 March 2018; Revised 24 August 2018; Accepted 08 October 2018
**Corresponding author: Mozayani@iust.ac.ir (N. Mozayani).*

Abstract
In this paper, we present a data-driven method for crowd simulation with holonification model. With this extra module, the accuracy of simulation will increase, and it generates more realistic behaviors of agents. First, we show how to use the concept of holon in crowd simulation and how effective it is. For this reason, we use simple rules for holonification. Using real-world data, we model the rules for joining each agent to a holon and leave it with random forests. Then we use this model in simulation. Also since we use data from a specific environment, we test the model in another environment. The results obtained show that the rules derived from the first environment exist in the second one. It confirms the generalization capabilities of the proposed method.
Keywords: *Crowd Simulation, Data-driven Model, Holonic Multi-agent Systems.*

1. Introduction

Crowd simulation is the simulation process of movement of a lot of objects or characters. High-quality crowd simulation in many applications such as entertainment (video and computer games), anomalies detection, virtual reality, education, and training is important. Crowd simulations can be divided into two groups according to the simulation goal: the first is a typical crowd such as the population of pedestrians, and the second category is the emergency evacuation.
One of the major applications of crowd simulation is in some computer games [1]. The most important requirements in this area, felt in these games, are collision avoidance and agent movement. In addition, the user expects any group that is in the game, to be as natural as possible and the agents in it to act like real people.
Simulation off the pedestrian population is used for traffic management in shopping centers. For example, by simulating a large store, we can find out how to arrange the ordering of items to minimize the average density of different parts of the store. In this case, customers will be more satisfied with their purchase. By so doing, the store will have the ability to be used by more people. However, the simulation of emergency evacuation means simulating the evacuation of people when an accident occurs for the building. This type of simulation helps designers build safer buildings and trainers to find better strategies for emergency exit planning.
Crowd simulation, like many problems, has some sub-problems that needs to be solved. The main steps in this type of simulation includes visualization and behaviors of the agents. The agent's behavior means any attribute that influences his actions, for example, how it goes toward the destination or avoids collision. In the recent years, much progress has been made in the field of visualization, and we can claim that the results in this area are very close to the reality. However, it is very difficult to model the crowd behavior in a way that is close to the reality. This is due to the complexity of the human behavior, especially when he is in a group. In addition, when a person feels a danger (emergency exit), his behavior will become different [2]. The main purpose of this paper is to provide a method for modeling crowd behavior.
Contribution: we propose a new holonic model for agent-based data-driven crowd simulation. We test this model with real-world scenarios. The

proposed method generates more realistic crowd behaviors than the compared method.

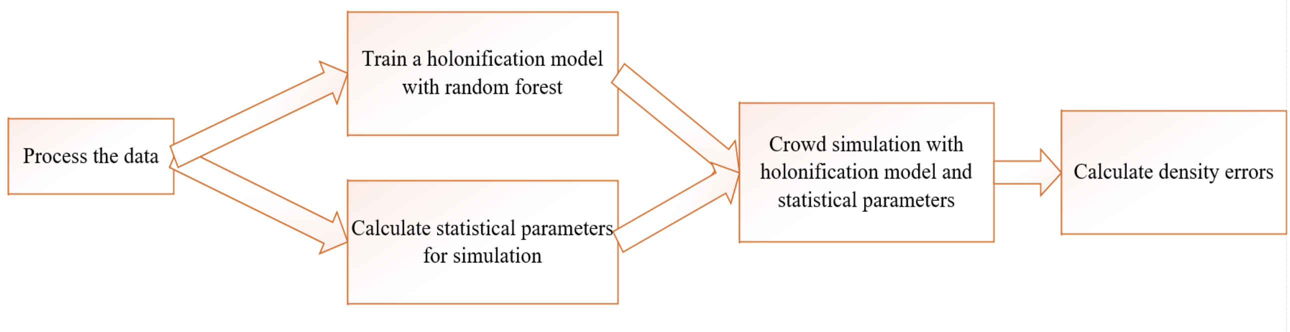

Figure 1. The main simulation procedure.

We also check the generalization of this model by training the holonification model with data of an environment and test the obtained model with data from another environment. We found that the proposed holonification method was robust to the environment data, which means that it has a good generalization ability.
The rest of the paper is organized as follows: Section 2: Related works; Section 3: Proposed method; Section 4: Evaluation and results; Section 5: Conclusions and future works.

2. Related works

For the first time, Reynolds created a major modification to the animation and crowd simulation [3]. In his experiments, it became clear that birds in a group of birds follow a few common behaviors such as staying close to the group and avoid collision. Therefore, the complex movement made by the birds is the result of a series of tasks that each bird does. The results obtained using this method showed that it was possible to simulate the complex behavior of birds with simple rules.
After this method, a lot of methods are introduced for crowd simulation. These methods can be divided into two categories: microscopic and macroscopic. In microscopic methods, attention is paid to the behavior of all agents in the system, while in macroscopic methods, global behaviors are considered. Some works are rule-based [4, 5], while others [6-12] use real-world data for simulation. Of course, some have also used the advantages of both [13]. This data is usually obtained by tracking people in the video of a crowd. With the advancements in tracking people in the field of computer vision, the dataset corresponds to tracking people in the crowd that can obtain accurately. To increase the reality of the simulation results, many models that are agent-based are suggested. Some of these models use psychological, social and physical parameters. In the recent years, the use of data-driven models has dominated other methods. There are some works for reducing the noise of the trajectories using interpolation [14]. This preprocessing can be useful for data-driven approaches.
In [15], the paths taken by individuals are extracted from the video, and then, using this information and based on the location of the agent, the velocity will be estimated. The problem with this method is that it works well only in a low-density population.
In [9], the proposed method uses human movement data. Before simulation, the examples are clustered and an ANN is trained to classify each sample to its cluster number. Thus at runtime to determine the velocity of the agent, the corresponding cluster is calculated. After that, the velocity of the agent is determined by similar samples in that cluster. More precisely, the search for similar samples is performed only within that cluster. This reduces the research time. Also in [16], the proposed method uses the neural networks for crowd simulation. This method uses real-world data for learning the motion of the human in the crowd by modeling it into an ANN. This method has a low test time but the training phase is expensive. Also the generalization of the model is not considered.
In [7], using the features of each agent, they are divided into two types, leader and follower. This method is presented to simulate a dense crowd. Its authors believe that the behavior of individuals is quite different when moving in a dense crowd. In this case, leaders attempt to find the best path to the destination, while followers follow the leaders with the same destination. In this method, the velocity of leaders is obtained from real-world data, while the followers' velocity is determined with some rules based on their leader, which is dynamically different at any time. There are some works for comparing different methods and data structures for finding the k-nearest neighbour that is used in crowd simulation [17]. The main focus of this works is to provide a near-optimal search framework. In [6], the required parameters are

extracted from the training set, which is the result of tracking people in the video of a crowd.

3. Proposed method

In this section, we introduce our method for crowd simulation with data. We used the dataset that was previously used in [6] to fairly validate the proposed method. For more information about the data, please refer to the original paper. The proposed model has a dual-layer structure, in which the top layer has high-level behaviours such as the arrival rate of people, arrival location, destination selection, and how to go there. In the bottom layer, there is a collision avoidance unit. In this model, a matrix containing velocity vectors is first obtained from the available data for each destination region. When simulating at any given moment, the agents enter the environment according to the arrival rate of the individuals and the possibility of entering from each input gate. Then, based on the probabilities available for each output gate, the destination of the agent is selected randomly according to its input gate and probabilities. In figure 1, the main procedures of this simulation are shown.

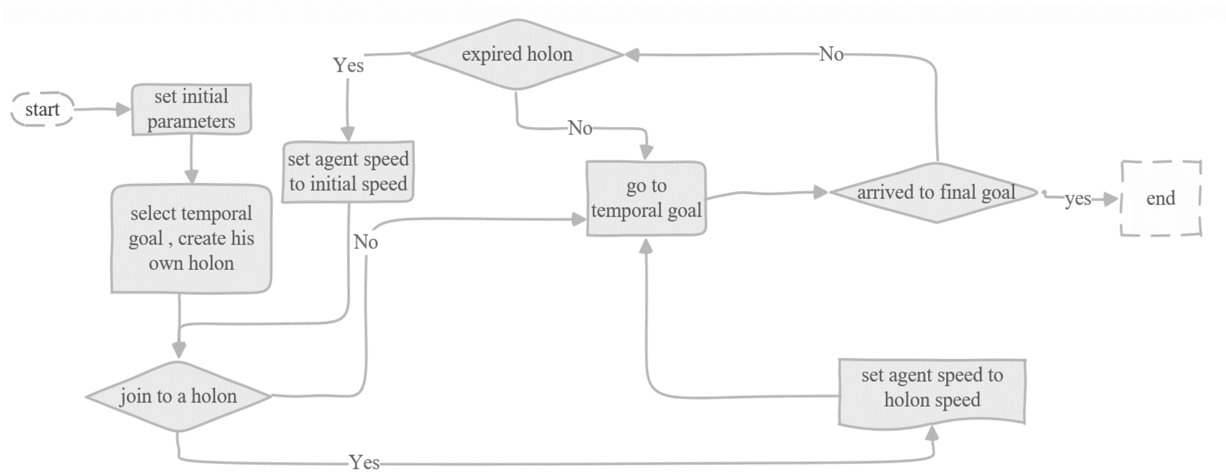

Figure 2. Simulation process for an agent.

The goal is to reconstruct the crowd for a specific environment that has its own training set. First, the source regions and destination regions are regarded. Each region can also be input and output simultaneously. Then with processing the data, it forms a matrix, each row representing an output gate and each column representing an input gate. Any value in this matrix indicates the probability of the corresponding output gate of that element for those who have entered the corresponding input port of that element. Also for a more accurate simulation, the arrival rate of agents from each gate should be specified. Before runtime, using the k-means clustering algorithm, the trajectories are clustered so that the trajectories in each cluster have the same output gate. Thus k must be equal to the number of outputs. For more convenience, discretization of the environment is done. Hence, the simulation environment is divided into cells of 1 m × 1 m. The velocities in each trajectory, which indicate movement in each cell, are used as the features for clustering. In fact, these features can be considered as a matrix. The value of cells that a person does not cross is zero. The center of each cluster is used at simulation time. In fact, these velocities show that how peoples whose destinations are similar move. This method has a good accuracy, and its results are closer to the real crowd compared to the other methods. One of the disadvantages of this method is that each agent moves independently from other agents, while in reality, there is no such independence. In many articles including [18, 19], this has been mentioned. In the proposed method, there is no such independence, and we show that by considering this dependency, it improves the accuracy and realization.

We already showed that the concept of holon can be used in data-driven crowd simulation [20]. That method is briefly explained in 3.1. In this paper, the method used to obtain rules for holonification is illustrated in 3.2.

3.1. Holonic model using rules

As mentioned in [7], in a crowd, the behaviour of each agent is dependent on the behaviours of other ones. First, we use simple rules for forming holon by agents. The conditions for being in the same holon for two agents are: 1. Having the same destination, and 2. The distance between them be below a threshold value (here is 5). If the conditions are satisfied by the agents, they will be in same holon.

Each agent is initially an independent holon. Then whenever the agent is closer to a holon whose destination is the same, it becomes a part of that holon. If these conditions are satisfied by two single-member holons, the agent that moves farther away from the destination will join another one. By joining the holon, the agent changes its speed to the holon's speed. The agent who creates that holon is selected as the supervisor of the holon. The speed of the holon is equal to the speed of its supervisor. This will eliminate a large number of collisions. As a result of collision reduction, the impact of collision avoidance unit on the simulation result is greatly reduced. The reason for this is that in the collision avoidance unit, the training data is not used and, in fact, it only optimizes the movements in order to avoid the collisions of the agents.

3.2. Holonic model using random forests

In the previous section, we used the simple rules for holonification. However, for more realization and reduction of the mean error, it would be better to specify a more precise law on how an agent joins the holon. This law should be obtained from the existing data. For this reason, we propose to create a model for holonification.

At this stage, we train a classifier to learn from the available data, under which conditions an agent joins a holon. The features we have used in this section include two agent's motion vectors and their speed. Each vector consists of two numbers that represent the direction of the agent in the x-, y-axis. To get this model, our choice is random forests [21]. Random forest is an ensemble of decision trees; which each decision tree uses only a subset of features. One of the most important advantages of this method is fast training and testing speed. It also has a very high generalization accuracy, and it is robust to noise.

3.2.1. Create dataset for holonification model

To obtain the samples in each frame, we look for agents that are close to each other. The closeness threshold is 5 m. After that, we get their distance in a certain number of later frames. For frame interval, we consider different numbers to determine the effect of this parameter. If these two agents still have a distance fewer than 5 m, after that particular frame, they will be labeled as positive, and otherwise, they will get a negative tag. Another situation will occur when at least one of the agents is absent after that particular frames. In this case, this sample will be ignored. We consider seven features for each sample. These features are described in table 1. In this description, V_i is the movement vector of agent i.

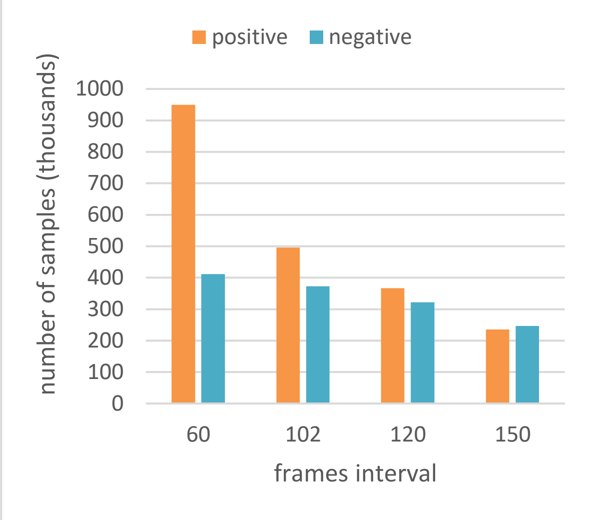

Figure 3. Comparison number of samples per class (first environment).

Table 1. Holonification model features.

Feature name	Description
v_{1x}	x-component of V_1
v_{1y}	y-component of V_1
v_{2x}	x-component of V_2
v_{2y}	y-component of V_2
s_1	The speed of the first agent in this time interval
s_2	The speed of the second agent in this time interval
c	The class label that indicates whether these agents stay together or not

3.2.2. Crowd simulation using a holonification model

After creating the dataset for the holonification model, we can train random forests. This model can determine whether two agents create a holon or not. The simulation process for an agent is shown in figure 2.

For an agent, if it joins a holon at a time, its speed will change as the supervisor of that holon. After a specific number of frames, the agent will decide again for being in the current holon or leaving it. If it leaves a holon, it can join another one. The only agent that cannot leave the holon is its supervisor.

3.3. Generality of model

We used a classifier to get the rules. This classifier, by capturing a number of attributes from the two agents mentioned above, indicates whether these two agents are in a holon. One of the shortcomings in this work is the generality. More precisely, it is possible that the obtained model be suitable only for this environment, and thus for another environment, another classifier is required to be fitted with features that are appropriate to the data in that environment. For this reason, we also check the generality of this model. To do this, we tested the model obtained from the second environment data in the first environment.

4. Evaluation and results

In this section, we first describe the evaluation method that is used. After that, the results of the proposed method will be shown and discussed.
We use 10-fold cross-validation for testing the results. For more details about this method, please refer to [6].

4.1. Evaluation method

We use local density [22] for evaluation. It is obtained from formula (1) for each cell, where p_i is the local density of cell with x-coordinate I and y-coordinate j. Also d (i, j, u) is the distance from the point u of the cell i, j, and E contains all the points of the paths and R is a parameter to normalize the number. The value for R is 2 m same as [6].

$$p_{i,j} = \frac{1}{\pi R^2} \sum_{u=1}^{E} \exp\left(-\frac{d(i,j,u)^2}{R^2} \right) \qquad (1)$$

$$p_{error} = \sqrt{\frac{\sum_{i=1}^{H} \sum_{j=1}^{W} \left(\bar{\rho}_{i,j} - \rho^0{}_{i,j} \right)^2}{H * W}} \qquad (2)$$

4.2. Holonification data

As mentioned in Section 3, for training the classifier, first the data should be generated. The class label for each sample is positive or negative. For a simpler comparison, the distribution for instances of each class is shown in figure 3 and 4. As one can see, the number of agents in the first environment is larger than the second one.

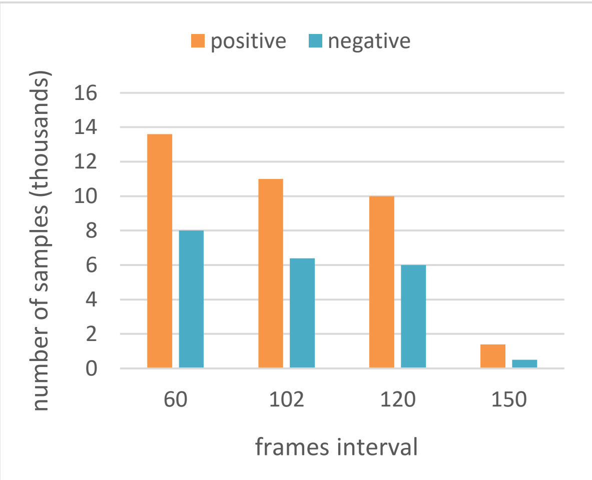

Figure 4. Comparison number of samples per class (second environment).

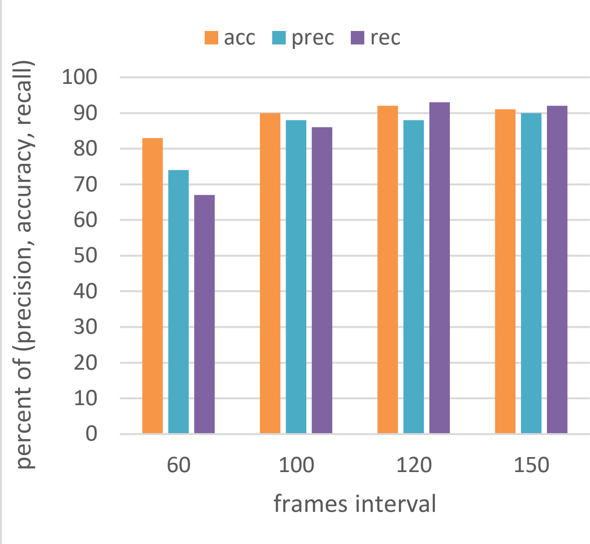

Figure 5. Accuracy, precision and recall for different frame interval.

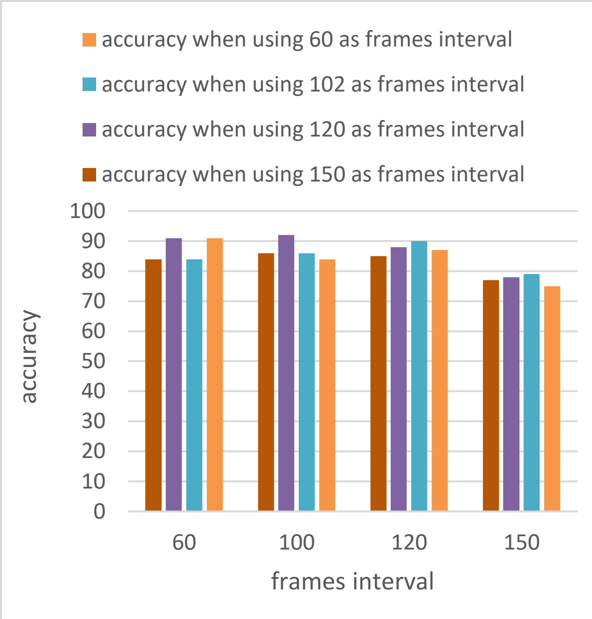

Figure 6. Accuracy for different frame interval, for making training set from second environment and test on the first environment.

We use the second environment just for testing the generality of holonification model and using creating holonification model by the first environment dataset.
In figures 3 and 4, by increasing the frame interval, the number of instances will decrease. The reason is that this dataset is obtained by tracking individuals and the chances of occlusion will increase over time.

4.3. Testing model

First, we validate the constructed model by cross-validation method with 10-fold. As mentioned above, the created data is imbalanced, especially when the frame interval is low. Hence, precision

and recall are obtained, in addition to the accuracy. The results obtained are shown in figure 5.
As it can be seen in figure 5, the accuracy of the model is above 83% in different frame intervals. Also the precision and recall are above 86% except when the frame interval value is 60. The reason is that when the frame interval value is 60, the data is imbalanced, as mentioned above. However, when the frame interval value is higher than 60, all measures are closer to each other and all are acceptable.

4.4. Checking generality of model

We also checked the generality of the model. The result is shown in figure 6. The horizontal axis shows the different frame intervals that are used for the test set. For each test set, the result is tested by different training sets generated from the second environment. As it can be seen in figure 6, the evaluating measures are acceptable. The evaluating measures are always above 75%. More precisely, both the environment and the frame interval do not play an essential role, which means that the holonification model is robust to the environment data and the frame interval. This confirms the generalization ability of the proposed method.

4.5. Simulation results

Finally, we test the proposed method as described. We call the proposed method "data-driven crowd simulation with holonification" (DCSH). The results are shown in table 2. Each method is tested ten times, where each time, p_{error} of the results is obtained by 10-fold cross-validation on training data. It can be observed that DCSH has better results than the compared method [6].
We believe that by reducing the collisions between the agents, the effect of collision avoidance module will decrease. The collision avoidance method use, in this simulation is RVO2 [23]. This method ignores the training data. Thus we believe that with decreasing the effect of collision avoidance module, the simulation result is more realistic.
The most advantage of the proposed holonification method is that it can be used temporarily during the simulation process. So, for each agent we use the holonification model for joining and leaving a holon.
As it can be seen in table 2, the proposed method performs better than the usual case, which is without holonification module and the local density error is less than the comparison method. This confirms our hypothesis that using the holonification model can make the simulation results more realistic.

Table 2. The p_{error} results.

Method	Mean (p_{error})	std (p_{error})
DCSH	273.6	8.3
D-ABC	332.1	14.2

4.6. Discussion

In the proposed method, the agents along the way temporarily form holons and change their velocity according to their corresponding holon. These changes cause a reduction in the number of collisions between the agents. When the number of collisions is reduced, the effects of the collision avoidance unit on the simulation results will decrease. Since the collision avoidance algorithm is not attentive to training data and its purpose is only to prevent the collisions, low impact of this unit enhances accuracy.
In addition, in the proposed method, the holonification is not dependent only on destinations of agents. This increases the reality in behaviors of the agents.
All the training and testing time of the holonification model is very fast due to the use of random forest. Thus using this module is very effective.
One of the limitations of the proposed method is that the complexity of holonification module is $O(n^2)$. We did not test the higher orders of holons. However, we believe that using higher orders of holons can increase the accuracy of holonification due to more similarity to the real crowd.

5. Conclusion and future works

One of the new and powerful concepts in the field of multi-agent systems is the holonic model. This model is used for applications such as reducing the amount of communication between agents and, consequently, reducing the use of communication resources. We used this concept for the first time in a data-driven crowd simulation method.
We first showed that the use of this concept in crowd simulation can increase the accuracy of the results obtained from the simulation. Then using the machine learning techniques, we found a model from the available data that modeled how the holon formed each agent. In fact, this model can determine whether the agent joins another holon at a moment or leaves the current holon.
We also showed that the model has a good generality. Thus after making the model from the data of the first environment, we tested the model by the data of the second environment and the accuracy was acceptable. Finally, using this model for helping the agent to join the holon or leave it,

to increase the realization of results, we decreased the total error.

There are some several works that can be done in the future. For making the results more realistic, multi-level holonification can be used. The agent's type can also be considered for providing more realistic results.

References

[1] Snape, J., Guy, S. J., van den Berg, J., Lin, M. C. & Manocha, D. (2012). Reciprocal collision avoidance and multi-agent navigation for video games. In Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence, pp. 49-52.

[2] Thalmann, D. (2007). Crowd simulation. Wiley Encyclopedia of Computer Science and Engineering.

[3] Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model , ACM, vol. 21, no. 4, pp. 25-34.

[4] Reynolds, C. W. (1999), March. Steering behaviors for autonomous characters. In Game developers conference, vol. 1999, pp. 763-782.

[5] Xiong, M., Lees, M., Cai, W., Zhou, S. and Low, M. Y. H., (2009). A rule-based motion planning for crowd simulation. In 2009 International Conference on CyberWorlds, IEEE, pp. 88-95.

[6] Zhong, J., Cai, W., Luo, L. & Zhao, M. (2016). Learning behavior patterns from video for agent-based crowd modeling and simulation. Autonomous Agents and Multi-Agent Systems, vol. 30, no. 5, pp.990-1019.

[7] Zhao, M., Zhong, J. & Cai, W. (2018). A Role-Dependent Data-Driven Approach for High-Density Crowd Behavior Modeling. ACM Transactions on Modeling and Computer Simulation (TOMACS), vol. 28, no. 4, p.28.

[8] Charalambous, P. & Chrysanthou, Y. (2014). The pag crowd: A graph based approach for efficient data-driven crowd simulation. In Computer Graphics Forum, vol. 33, no. 8, pp. 95-108.

[9] Zhao, M., Turner, S. J. & Cai, W. (2013). A data-driven crowd simulation model based on clustering and classification. In Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications, IEEE Computer Society, pp. 125-134.

[10] Bera, A., Kim, S. & Manocha, D. (2016). Online parameter learning for data-driven crowd simulation and content generation. Computers & Graphics, vol. 55, pp.68-79.

[11] Karamouzas, I., Sohre, N., Narain, R. & Guy, S. J. (2017). Implicit crowds: Optimization integrator for robust crowd simulation. ACM Transactions on Graphics (TOG), vol. 36, no. 4, p.136.

[12] Tatari, F. & Naghibi-Sistani, M. B. (2015). Optimal adaptive leader-follower consensus of linear multi-agent systems: Known and unknown dynamics. Journal of AI and Data mining, vol. 3, no. 1, pp.101-111.

[13] Zhao, M. (2015). A clustering based approach for realistic and efficient data-driven crowd simulation. arXiv preprint arXiv:1506.04480.

[14] Qiao, G., Yoon, S., Kapadia, M. & Pavlovic, V. (2018), April. The Role of Data-Driven Priors in Multi-Agent Crowd Trajectory Estimation. In Thirty-Second AAAI Conference on Artificial Intelligence.

[15] Musse, S. R., Jung, C. R., Jacques Jr, J. C. & Braun, A. (2007). Using computer vision to simulate the motion of virtual agents. Computer Animation and Virtual Worlds, vol. 18, no. 2, pp.83-93.

[16] Wei, X., Lu, W., Zhu, L. & Xing, W. (2018). Learning motion rules from real data: Neural network for crowd simulation. Neurocomputing, vol. 310, pp.125-134.

[17] Vermeulen, J. L., Hillebrand, A. & Geraerts, R. (2017). A comparative study of k-nearest neighbour techniques in crowd simulation. Computer Animation and Virtual Worlds, vol. 28(3-4), p.e1775.

[18] Pelechano, N. & Badler, N .I. (2006). Improving the realism of agent movement for high density crowd simulation. University of Pennsylvania, Center for Human Modeling and Simulation.

[19] Karamouzas, I. & Overmars, M. (2010). Simulating the local behaviour of small pedestrian groups. In Proceedings of the 17th acm symposium on virtual reality software and technology, ACM, pp. 183-190.

[20] Peymanfard, J. & Mozayani, N. (2018). A Holonification Model Using Random Forest for Crowd Simulation. In 2018 9th International Symposium on Telecommunications (IST), IEEE, pp. 311-314.

[21] Breiman, L. (2001). Random forests. Machine learning, vol. 45, no. 1, pp.5-32.

[22] Helbing, D., Johansson, A. & Al-Abideen, H. Z. (2007). Dynamics of crowd disasters: An empirical study. Physical review E, vol. 75, no. 4, p.046109.

[23] Van den Berg, J., Lin, M. & Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In 2008 IEEE International Conference on Robotics and Automation, IEEE, pp. 1928-1935.

نشریه هوش مصنوعی و داده کاوی

ارائه روشی داده محور برای شبیه‌سازی ازدحام با استفاده از مدل هولون‌سازی

جواد پیمان فرد و ناصر مزینی*

گروه مهندسی کامپیوتر، دانشگاه علم و صنعت، تهران، ایران.

ارسال ۲۰۱۸/۰۳/۱۳؛ بازنگری ۲۰۱۸/۰۸/۲۴؛ پذیرش ۲۰۱۸/۱۰/۰۶

چکیده:

در این مقاله روشی داده محور برای شبیه‌سازی ازدحام ارائه داده‌ایم که در آن با کمک مفهوم هولون توانسته‌ایم دقت شبیه‌سازی را افزایش داده و خروجی آن را به یک ازدحام واقعی نزدیک‌تر کنیم. برای این کار ابتدا نشان داده‌ایم که می‌توان از مفهوم هولون در مسئله‌ی شبیه‌سازی ازدحام استفاده کرد و این نوع نگاه به جمعیت، سبب افزایش دقت کار می‌شود. سپس با استفاده از داده‌های دنیای واقعی، قواعد پیوستن هر عامل به یک هولون و خروج از آن را به کمک یک رده‌بند، مدل کرده و از آن مدل در شبیه‌سازی کمک گرفته‌ایم. همچنین به دلیل اینکه از داده‌های مربوط به یک محیط در این راستا استفاده کردیم، مدل به‌دست‌آمده را با داده‌های مربوط به یک محیط دیگر آزمودیم و مشخص شد که قواعد به‌دست‌آمده از محیط اول تا حد زیادی در محیط دوم نیز وجود دارد. این موضوع مهر تأییدی بر عمومیت و جامعیت روش پیشنهادی می‌باشد.

کلمات کلیدی: شبیه‌سازی ازدحام، مدل داده محور، سیستم‌های چندعاملی هولونی.