

Research Article

An Efficiency Control Method Based on SFSM for Massive Crowd Rendering

Lei Lyu ^{1,2}, Jinling Zhang,³ and Meilin Fan¹

¹School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

²Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan 250358, China

³School of Information, Renmin University of China, Beijing 100872, China

Correspondence should be addressed to Lei Lyu; lvbu007@163.com

Received 11 March 2018; Accepted 5 September 2018; Published 1 October 2018

Academic Editor: Constantine Kotropoulos

Copyright © 2018 Lei Lyu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For massive crowds, users often have the need for interactive roaming. A good roaming effect can make the user feel immersed in the crowd, and the scenes need to be populated with crowds of people that make the environment both alive and believable. This paper proposes a method of efficiency control for massive crowd rendering. First, we devise a state machine mechanism based on self-feedback, which can dynamically adjust the accuracy of crowd model rendering according to the relationship between the speed of the system rendering and the speed the users expect. Second, we propose a movement frequency update method to perform the frequency of motion update based on the distance between the individual and the viewpoint. In addition, we propose a variable precision point sampling drawing strategy to render the individual with different sampling precision. The state machine system in this paper effectively integrates two core technologies for dynamically controlling the accuracy of the model, ensuring visual efficiency, improving the rendering efficiency, and satisfying the fluency of users' roaming interaction.

1. Introduction

The simulation of massive crowds is important in many fields of computer graphics, including real-time applications such as games, as crowds can make otherwise static scenes more realistic and enhance believability. The existing massive crowd rendering methods often use the following three types of techniques to improve the rendering speed: the level of detail (LoD) technique [1, 2], the image-based rendering (IBR) technique [3, 4], and the point sample rendering (PSR) technique [5, 6].

In the LoD method, the original grid is simplified to varying degrees according to the distance from the drawing object to the viewpoint to reduce the total drawing figure. The IBR technique has recently grown in popularity [7]. This is because it allows extremely fast rendering of only one picture for each agent and can render the impostors automatically. The drawback is that the stored images are highly correlated with the consumption of texture memory. In addition, interpolating the images is time-consuming and generally

prohibitive within a given time range. In the PSR technique, there is no need to store and maintain consistent topological information. Thus, it is more flexible than triangle meshes. However, there are several limitations to this technique. For example, the points become independent from the original mesh, and loading animations becomes difficult if the point samples are the result of decimating the mesh for LoD.

The previous research either simplifies the complex models of original triangular patches according to the changes in users' perspectives or replaces simple triangles with simple graph elements. However, when the user's perspective changes dynamically, it may cause a mutation in the rendering efficiency. In some practical applications, users often need to observe from multiple locations and perspectives because of the large area of the group distribution. The existing group rendering methods differ greatly in the speed of single frame rendering due to changes in the rendering task under different perspectives. This creates the fluent online roaming experience for the users. Therefore, there is an urgent need to control the efficiency of massive crowd rendering.

This paper proposes an efficient control method for massive crowd rendering. In our method, we devise a self-feedback state machine (SFSM) to control efficiency. It contains three states, and each state corresponds to a rendering precision. The state is determined by the time cost of rendering a frame. Thus, the rendering speed can be kept within a reasonable range, achieving a real-time effect.

Due to the continuity of the group movement and changing views, we adopt the rendering time cost of the previous frame as the feedback information to predict the possible time for the current frame. It is also used to determine the state of the SFSM. Therefore, when the rendering speed of a massive group scene is different from the user's expected speed, the SFSM can automatically detect the abnormality and quickly adjust it to the desired speed. Additionally, we save time consumption by reducing the update frequency of individuals who are far from the point of view. These individuals are in a visually insensitive region, so they have little effect on the overall visual effect. To further reduce the total number of drawing tasks, the variable precision sampling technique uses a relatively rough point sampling model to replace a relatively fine model, which is similar to reducing the overall resolution of the screen.

Experiments show that our efficiency control method can stabilize the rendering efficiency within the user's expectations to within 0.6 s for the scenery drawing of tens of thousands of sports fans. The speed deviation is controlled within 10 ms.

2. Related Work

Regarding massive crowd rendering, there has been a large amount of work in this field [8–12]. These studies have provided a complete, up-to-date overview of the state of the art in crowd rendering, with an in-depth comparison of techniques in terms of quality, resources, and performance. This section presents a brief overview of related work on massive crowd rendering.

LoD is an approach that has been used to improve the performance of real-time crowd rendering [13]. The idea is to replace small, distant, or unimportant objects in the scene with an approximate simulation model when a crowd is being drawn. At each frame, the system selects the appropriate model or resolution according to the model's distance to the viewpoint [14]. Brogan and Hodgins [15] adopted the LoD technique to control the motions of massive crowds. They simulated massive crowds by dynamically switching between these LoDs based on a simplified version of a physically simulated character as a simulation LoD. O'Sullivan *et al.* [16] proposed a method combining several levels of detail for massive crowds. However, this generated poor results when making use of low-resolution meshes because too much detail is removed. Animation artifacts due to the loss of joint vertices can also occur, thus reducing the overall visual realism of the virtual human. Additionally, it has been found that a low-resolution model is not perceptually equivalent to its high-resolution counterpart at conveying subtle variations in motion [17], illustrating the importance of accounting for animation when selecting LoD schemes. Billboard clouds

presented an appealing alternative to extreme geometric LoD simplification [18].

Rendering impostors instead of geometry has proven to greatly improve rendering efficiency but will lead to visual appearance. There is a large amount of research regarding IBR systems. Visual impostors introduce image-based techniques into a geometrically rendered environment. The idea is to replace parts of the models with an image representation textured onto a simple, normally quad, geometric shape. In this way, the rendering time of an object is reduced to a constant texturing overhead. Tecchia [19] used the IBR technique to render massive crowds in simulating a virtual urban environment. First, a number of possible views for a human model are prerendered and stored in memory. Then, the closest view from the set is used to display the character during the simulation. Likewise, Dobbyn *et al.* [3] introduced a new approach called Geopostors, in which the detailed geometries and impostors are combined to generate virtual humans. They constructed Geopostors by mingling many image maps produced by normal maps, detail maps, and a set of customizable color materials. Geopostors achieved interactive frame rates and visually realistic simulations with large-scale agents.

Using points as a new primitive to render geometry was suggested early in Levoy and Witted's report [20]. Its idea is to render a model using a mass of points. A Gaussian filter or surface splatting [21] can be performed to fill in the possible gaps. A survey on point-based techniques was proposed by Kobelt and Botsch [22]. In [23], the authors proposed using only point-based models to replace models that are far away from the viewpoint. Thus, PSR is more useful and faster when the triangles of a model cover a pixel or less and do not need to store and maintain globally consistent topological information. Therefore, they are more flexible compared to triangle meshes. Nevertheless, this technique has some limitations. For example, it easily leads to independence from the initial model, and it is difficult to load animations if the point samples are the result of decimating the mesh for LoD.

3. Efficiency Control Method

To achieve high performance when rendering thousands of agents, this paper presents an efficiency control strategy. The idea is to adjust the precision of the models by the SFSM during rendering. When the rendering speed is lower than the users' expectations, the rough-precision model is used instead of the fine-precision model to increase the rendering speed. By contrast, when the rendering speed meets the users' expectations, fine-precision models are used to ensure better visual quality. At the same time, a movement updating method and a variable precision sampling method are also used to achieve the dynamic adjustment of the model rendering and to minimize the loss of visual effects.

Figure 1 shows the general process of crowd rendering. Most of the time is spent on the movement (position, orientation, action, etc.) updating at each frame and the rendering of the models. Addressing these problems, the goal of our technology is variable frequency motion updating and variable precision sampling methods.

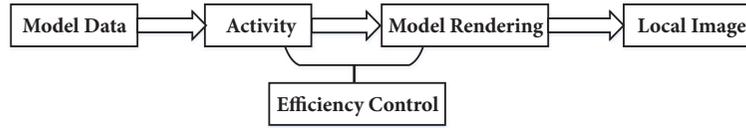


FIGURE 1: Group rendering process.

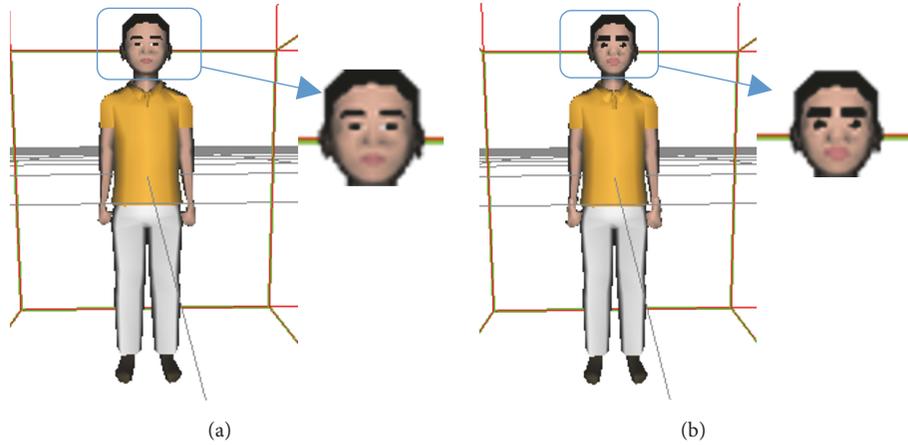


FIGURE 2: Model points sampling effect. (a) 1246 triangles. (b) 946 triangles, 7773 sampling points.

The principles and implementation of the two algorithms will be described in Section 3.1 and Section 3.2, respectively, and the structure of the SFSM and its efficiency control process will be described in Section 3.3.

3.1. Variable Frequency Movement Update. Rachel et al. [17] reported the distances from the camera at which impostors and different simplified meshes are perceptually equivalent to high-resolution geometry. Our method for variable frequency movement updating is based on this fact.

Our purpose for controlling the rendering efficiency is achieved by reducing the frequency of the movement update of the individuals who are far away from the camera.

We use f to represent the update frequency of the crowd movement. A movement update is performed every f frame. Here, f is an integer greater than or equal to 1 and is calculated as follows:

$$f = \begin{cases} 1 & \text{if } d_{cur} < d \\ (d_{cur} - d) \times \theta & \text{if } d_{cur} \geq d \end{cases} \quad (1)$$

where d_{cur} is the distance along the viewing direction from the camera to the virtual human model. We will reduce the update frequency only when the d_{cur} is greater than d . d is determined by the SFSM. θ is a constant, and its value is between 0.05 and 0.1 based on the experience.

3.2. Variable Precision Sampling Rendering. Because the PSR method has the advantage of drawing effect and rendering group size in real-time and can recognize the rapid drawing of millions of people on a single PC [5], we propose a variable precision sampling technology based on PSR technology. It

realizes the dynamic control of the rendering efficiency by using the multiprecision point sampling model.

3.2.1. Massive Crowd Rendering Based on Point Sampling. To achieve the rapid rendering of the three-dimensional model, the principle of drawing technology based on point sampling is to use the projection area of the corresponding point element to replace the original triangular facet elements. We successfully applied the technology to simplify the virtual human model [5] (see Figure 2) and realized crowd rendering with 30,000 to 50,000 agents (Figure 3).

The group mapping method based on point sampling can effectively improve the rendering efficiency of group scenes and the scale of real-time rendering. However, as with other group mapping methods, the basic idea of the point rendering method is to dynamically simplify the model. When the user's view changes, there may be changes in the speed of the screen update, resulting in a nonsmooth visual experience.

3.2.2. Variable Precision Sampling Technology. In the original point sampling, the principle is to use a projection area that is large enough to replace the triangular facet element that is the drawing model to save time. When the size of the pixel is increased, the projection of the pixel is also increased. For the same model, the number of pixels is increased, and the number of triangular facets is reduced, so the drawing speed is increased. On the other hand, the triangular facets are increased, which results in a more elaborate model, and the drawing speed is reduced.

Point-based techniques were proposed early by Levoy and Witted's [20], and they suggested the use of points as a new

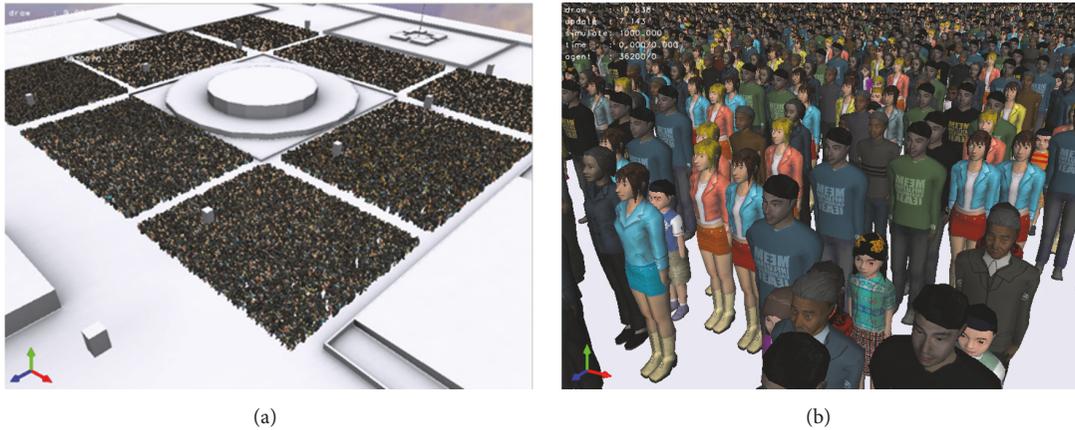


FIGURE 3: Our massive crowd simulation system. (a) The full view of the scene, the scene contains a total of 36200 individuals. (b) The local magnification of the scene.



FIGURE 4: Rendering effect of different sampling density.

primitive to render geometry. The idea is to render a surface using a vast number of points. Point-sampled objects do not need to store and maintain globally consistent topological information. Therefore, they are more flexible compared to triangle meshes.

The principle of our variable precision sampling is to adjust the size of the sampling point dynamically by increasing (or decreasing) the projection area of the sampling point screen, according to the requirement of the rendering speed. The effect of dynamically modifying the size of the sampling point is equivalent to a change in screen resolution (Figure 4). For each frame, the size of the sampling point is set by the SFSM.

The two figures on the first line in Figure 4 show that the sampling point weights are close to 0.2 pixels and the

frame rate is 21.3 *fps* when we use a higher precision to render the group effect. The other two figures on the second line in Figure 4 are the low-precision rendering effect, whose sampling point weights are close to 1 pixel, and the rendering frame rate is 65.5 *fps*.

3.3. Self-Feedback State Machine. Based on the above two methods, we design the SFSM to automatically control the rendering efficiency for crowd movement. The SFSM dynamically monitors the time overhead drawn by each frame and compares it with the drawing speed expected by the user who determines whether the rendering speed of the current system reaches the level of user satisfaction and what kind of acceleration strategy and precision should be used to draw the current frame.

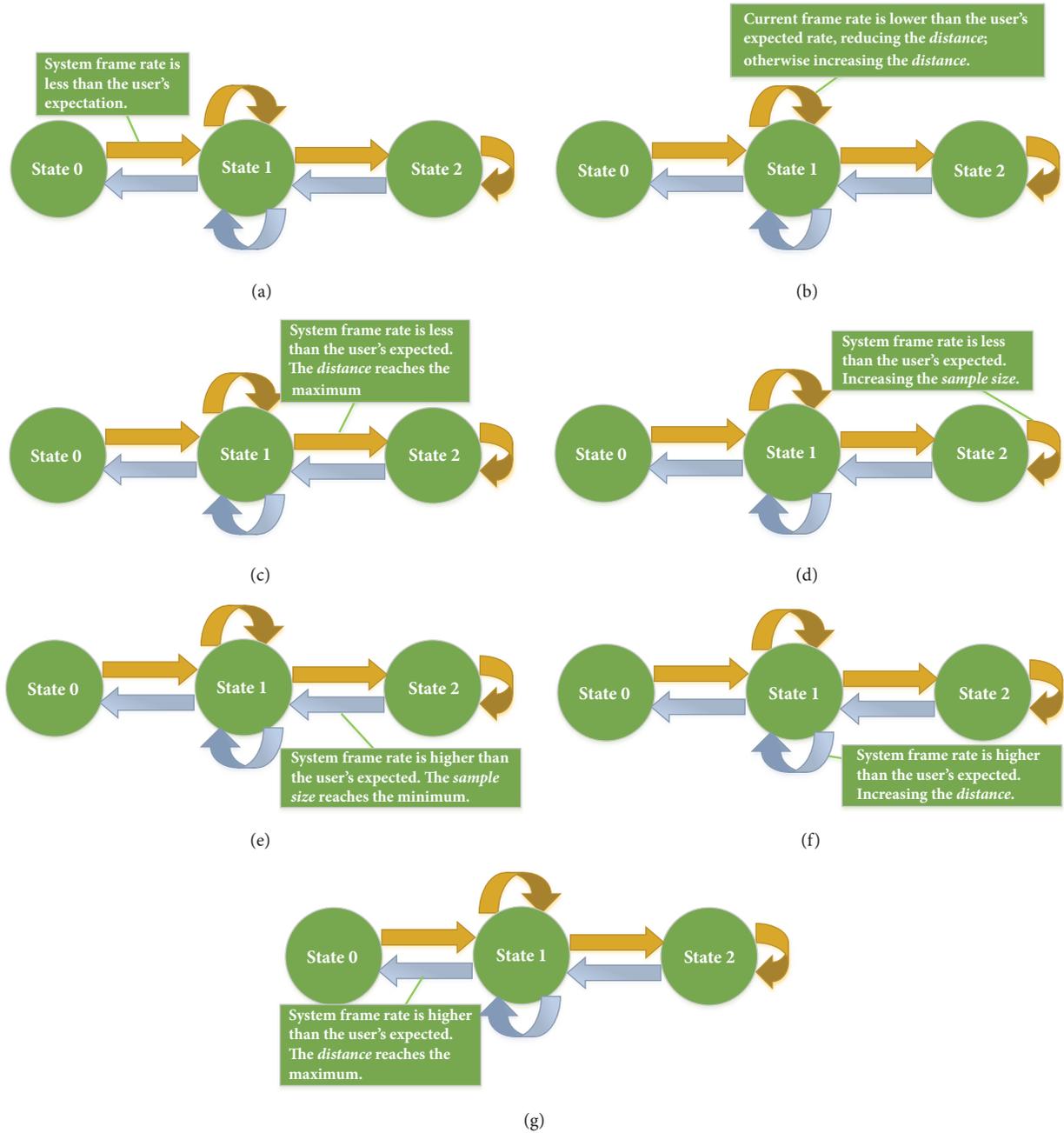


FIGURE 5: State transition mechanism of state machine.

There are three rendering states in the SFSM: State 0, State 1, and State 2, corresponding to different acceleration strategies. In State 0, there is no movement update operation with variable frequency. The model is rendered by the original model. It is the most detailed drawing, and the time cost is relatively large. In State 1, the variable frequency movement update needs to be performed, and the sampling point size remains constant in the process. In State 2, the variable frequency movement update must also be performed, and each individual is rendered by a variable precision point. In this state, the rendering speed is the fastest.

Figure 5 shows the transition among the states, where d is a variable in State 1 (corresponding to the d in (1)). It means that the individual whose distance to the camera is greater than d begins to perform the variable frequency movement update. The larger the distance, the slower the update frequency.

The distance is obtained by the self-learning method. Suppose that the time cost of the i -th frame is T_i and the corresponding distance weight is D_i . We assume that $(D_i - D_{i-1})$ is ΔD_i and the time cost expected by users is t , and in the current view, the farthest distance visible is L . When

the current frame rate of the system does not reach the users expected frame rate (i.e., $T_i < t$),

$$D_i = \begin{cases} L & (i = 0) \\ \frac{5}{6L} & (i = 1) \\ \frac{t - T_{i-2}}{T_{i-1} - T_{i-2}} \times D_{i-1} - \frac{t - T_{i-1}}{T_{i-1} - T_{i-2}} \times D_{i-2} & (i > 1) \end{cases} \quad (2)$$

In the initial state, the weights are specified by the user. D_1 size is L . T_1 is the time cost of the first frame (according to the definition of the state transition condition, $T_1 > t$ when the system is in State 1), D_2 is $5/6L$, and T_2 is the time cost of the second frame. The derivation process of formula (2) is a systematic self-learning process. The steps are as follows.

When $T_i > t$, then

$$\Delta D_i = \Delta D_{i-1} \times \frac{t - T_{i-1}}{T_{i-1} - T_{i-2}} \quad (i > 2) \quad (3)$$

Further conversion is as follows:

$$D_i - D_{i-1} = (D_{i-1} - D_{i-2}) \times \frac{t - T_{i-1}}{T_{i-1} - T_{i-2}} \quad (i > 2) \quad (4)$$

In the learning process, if $T_i < t$, there is an ‘‘over learning condition’’, then

$$\Delta D_i = \frac{-1}{2} \Delta D_{i-1} \quad (5)$$

By conversion,

$$D_i = \frac{1}{2} D_{i-1} + \frac{1}{2} D_{i-2} \quad (6)$$

where D_i is set to a constant $(11/12) * L$ when $i = 2$.

In the self-learning process, ΔD_i gradually converges, and when it is less than a certain weight h (h is usually set between 10 and 20 meters), ΔD_i is set to a constant and the size is h . When $D_i < (2/3) * L$, the SFSM will transition to State 2 from State 1.

When the system is in State 2, the parameter sample size denotes the size of the sampling points. If the rendering speed is lower than the speed the user can tolerate, the system switches from a lower state to a higher state (Figures 5(a)–5(c)), and then the efficiency of the crowd rendering will increase. Conversely, it will be reversed in the reverse direction (Figures 5(d)–5(g)), and the accuracy of the model will decrease.

The weight of the sample size is controlled by the linear change. Due to the use of the original model of the triangular drawing, the sampling point weight is 0 in the initial state. We set the i -th frame sampling point weight as S_i according to the dynamic increase or decrease of frame rate demand. The specific method is as follows:

$$S_i = \begin{cases} S_{i-1} + \tau & (T_{i-1} > t) \\ S_{i-1} - \tau & (T_{i-1} < t) \end{cases} \quad (1 < \tau < 2) \quad (7)$$

4. Experimental Results

We measured the run-time performance of our efficiency control method using an I7 3.4 GHz processor with 8 GB RAM and a GeForce 1050 graphics card with 2G of video memory. The original model of the movement individual contains 1000 patches.

To verify the response ability against the changes in the efficiency control during rendering, in our first test (see Figure 6), we rendered a crowd with twenty thousand individuals in a static perspective on a single PC. In the initial state, the speed of the user’s expectation is 1000 *ms/frame*, and the system is in State 0. In the 30th frame, the speed of the user’s expectation is adjusted to 300 (± 10) *ms/frame*, the system state is transformed into State 1 (see Figure 7), and in this state, the time cost for each frame is between 300 *ms* and 310 *ms*. In the 100th frame, the speed of the user’s expectation is adjusted to approximately 95 (± 10) *ms/frame*, and the system state is transformed into State 2 (see Figure 8). From these figures, we can see that, in the roaming process, the rendering efficiency of the user’s expectation has changed twice, and the time cost for each frame remains between 300 *ms* and 310 *ms* after the 100th frame.

Figure 9 clearly illustrates the performance of crowd rendering with and without efficiency control. There are two state adjustments in the 30th and 100th frames, and the rendering speed for both increased. We also find that the average time cost between two adjustments is 0.55 s, and the control rate in efficiency reaches 96.6%.

From this experiment, we can see that our approach can quickly respond to requirements for changing speed and control the speed of massive group scenes within the range of users’ expectations.

In our next experiment, to verify the ability to respond to changes in the rendering viewpoint, we also perform a crowd simulation with 20,000 individuals on a single PC. The only difference is that this simulation is in the dynamic perspective. We render the animation from three viewpoints: close range view, medium range view, and remote range view, and the rendering effect is shown in Figure 10. At rendering time, the speed of the user’s expectation is 95 *ms/frame* in the initial state, and the system is in State 1. From this state, the roaming view will make a change periodically. At the same time, the system will adjust the rendering precision by SFSM to ensure the smoothness of the screen.

In this experiment, we took advantage of both movement updating and variable precision sampling. The resulting time cost for each frame is reported in Figure 11, where the time costs of each frame with and without efficiency control are represented in red and blue lines, respectively. We can see that the rendering speed with our system can be maintained in a stable range. The control rate in efficiency is up to 97.8% using the SFSM and is higher than that in the case of no efficiency control at 69.7%. This indicates that our method can respond quickly to the user’s dynamic changing perspective and control the rendering speed in a desired range, guaranteeing the user roaming visual fluency in the case of perspective changes.

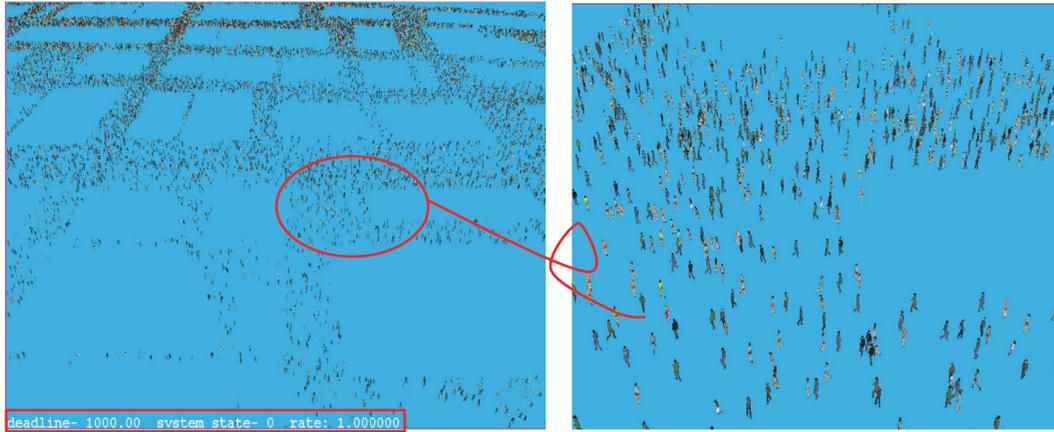


FIGURE 6: Rendering effect in State 0. The left picture shows that the group drawing vision is in State 0 and the drawing frame rate is 1.0 fps; the right picture shows the partial magnification. And we can see that the model accuracy is relatively high.



FIGURE 7: Rendering effect in State 1. The left picture shows that the group drawing vision is in State 1 and the drawing frame rate is 3.3 fps; the right picture shows the partial magnification. And we can see the model accuracy is remained in State 0.



FIGURE 8: Rendering effect in State 2. The left picture shows that the group drawing vision is in State 2 and the drawing frame rate is 10.5fps; the right picture shows the partial magnification and we can see the model accuracy is slightly lower than the State 0, but the fineness of the model is not noticeable in the vision.

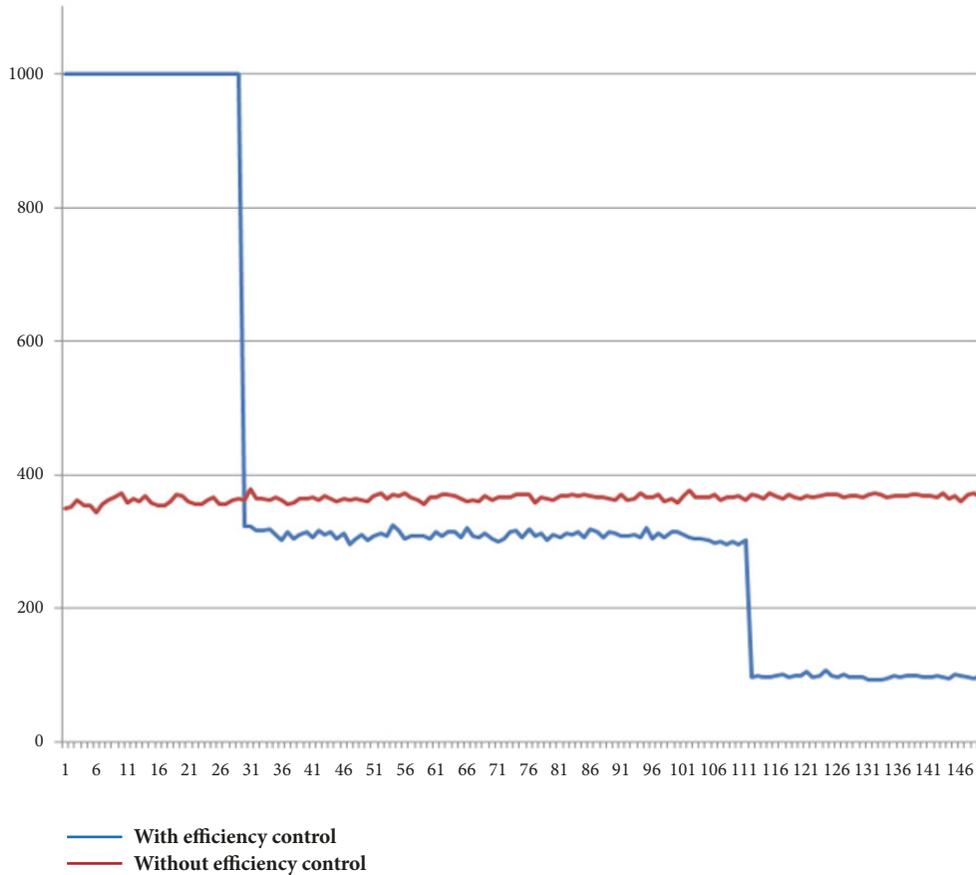


FIGURE 9: Efficiency curve of the first experiment. Vertical axis indicates the time cost (ms); the horizontal axis indicates the animation frame.

The two experiments show that our efficiency control mechanism provided for crowd rendering can effectively control the efficiency of massive crowd rendering and provide users with a smoother visual experience. The control rate in efficiency is relatively high, the frame rate of the system at rendering time can satisfy the user's requirements, and the response time is shorter when the users' expected frame rate changes dynamically.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Additional Points

Summary and Prospect. In this paper, we proposed a method to control the efficiency of rendering massive crowds and designed a precision control SFSM. Through dynamic adjustment of the mapping accuracy of the population model, the dynamic control of the rendering efficiency is realized, and the scene of large-scale group movement guarantees smooth roaming. Our method has several advantages. First, our proposed multilevel SFSM method can automatically respond to changes in task and user needs, and it can also quickly and effectively adjust the scene rendering speed and

stability within a range to ensure the user's online roaming fluency. Second, under the premise of ensuring rendering efficiency, multilevel SFSM uses variable frequency motion updating technology and variable precision point sampling technology to reduce the loss of visual effects.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Lei Lyu designed the concept of this paper and wrote the original manuscript; Jinling Zhang made some comments about this study and revised the manuscript; Meilin Fan conceived and designed the experiments. All authors have read and approved the final manuscript.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (nos. 61502505, 61472232, and 61572299), the Shandong Key Research and Development Program (no. 2017GSF20105), Major Program of Shandong Province Natural Science Foundation (no. ZR2018ZB0419), Natural Science Foundation of Shandong Province (no.

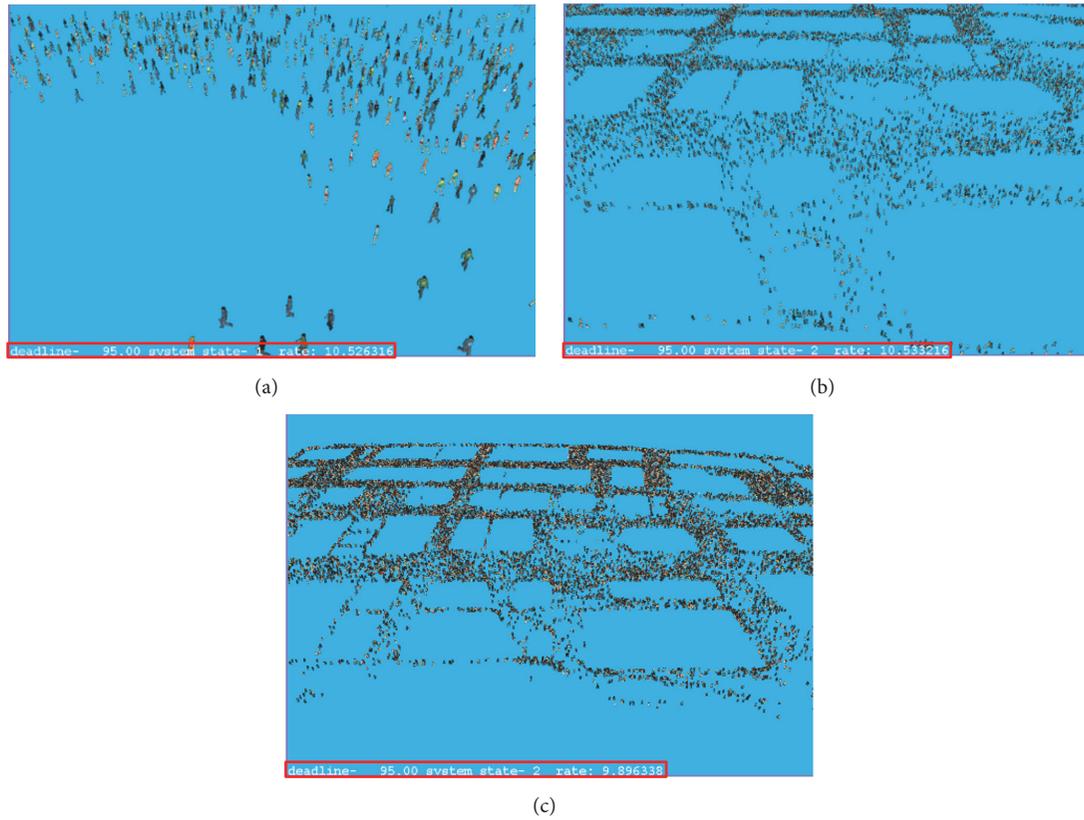


FIGURE 10: Scenes with three different viewpoints. (a) Close range view. (b) Medium range view. (c) Remote range view.

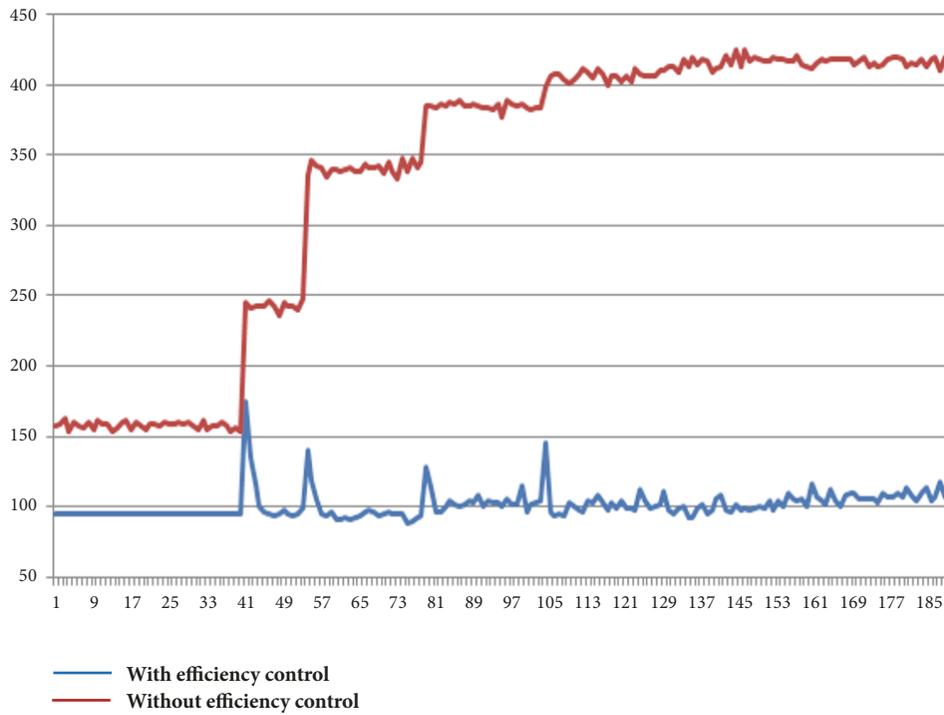


FIGURE 11: Efficiency curve of the second experiment. Vertical axis indicates the time cost (ms); the horizontal axis indicates the animation frame.

ZR2016FB13), and Shandong Province Higher Educational Science and Technology Program (no. J16LN09).

References

- [1] S. Kircher and M. Garland, "Progressive multiresolution meshes for deforming surfaces," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '05)*, pp. 191–200, July 2005.
- [2] E. Landreneau and S. Schaefer, "Simplification of articulated meshes," *Computer Graphics Forum*, vol. 28, no. 2, pp. 347–353, 2009.
- [3] S. Dobbyn, J. Hamill, K. O'Connor, and C. O'Sullivan, "Geopostors: A real-time geometry / impostor crowd rendering system," in *Proceedings of the I3D 2005: ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 95–102, USA, April 2005.
- [4] L. Kavan, S. Dobbyn, S. Collins, J. Žára, and C. O'Sullivan, "Poly-postors: 2D polygonal impostors for 3D crowds," in *Proceedings of the Symposium on Interactive 3D Graphics and Games, I3D 2008*, pp. 149–155, USA, February 2008.
- [5] B. Wang and N. Holzschuch, "Point-Based Rendering for Participating Media with Refractive Boundaries," *IEEE Transactions on Visualization and Computer Graphics*, vol. 99, pp. 1–14, 2017.
- [6] B. Chen and X. P. Nguyen M, "A Hybrid Point and Polygon Rendering System for Large Data," in *Proceedings of the Conference on Visualization. IEEE Computer Society*, pp. 45–52, 2001.
- [7] F. Tecchia and Y. Chrysanthou, "Real-Time Rendering of Densely Populated Urban Environments," in *Rendering Techniques 2000*, Eurographics, pp. 83–88, Springer Vienna, Vienna, 2000.
- [8] G. Ryder and A. M. Day, "Survey of real-time rendering techniques for crowds," *Computer Graphics Forum*, vol. 24, no. 2, pp. 203–215, 2005.
- [9] M. A. Azahar, M. S. Sunar, D. Daman, and A. Bade, "Survey on Real-Time Crowds Simulation," in *Technologies for E-Learning and Digital Entertainment*, vol. 5093 of *Lecture Notes in Computer Science*, pp. 573–580, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [10] N. Pelechano, N. Badler, and J. Allbeck, "Virtual Crowds: Methods, Simulation, and Control," *Synthesis Lectures on Computer Graphics Animation*, vol. 3, no. 2, 2008.
- [11] D. Thalmann and S. R. Musse, *Crowd Simulation*, Springer London, London, 2013.
- [12] A. Beacco, N. Pelechano, and C. Andújar, "A Survey of Real-Time Crowd Rendering," *Computer Graphics Forum*, vol. 35, no. 8, pp. 32–50, 2016.
- [13] B. Ulicny, P. De Heras Ciechomski, and D. Thalmann, *Crowd-brush: Interactive Authoring of Real-Time Crowd Scenes*, OAI, 2014.
- [14] D. Luebke, B. Watson, and J. Cohen, *Level of Detail for 3D Computer Graphics*, 2003.
- [15] D. C. Brogan and J. K. Hodgins, "Simulation level of detail for multiagent control," in *Proceedings of the 1st International Joint Conference on: Autonomous Agents and Multiagent Systems*, pp. 199–206, Italy, July 2002.
- [16] C. O'Sullivan, J. Cassell, H. Vilhjálmsson et al., "Levels of detail for crowds and groups," *Computer Graphics Forum*, vol. 21, no. 4, pp. 733–741, 2002.
- [17] R. McDonnell, S. Dobbyn, and C. Sullivan O, "LOD human representations: A comparative study," in *Proceedings of the First International Workshop on Crowd Simulation*, 2005.
- [18] X. Décoret, F. Durand, F. X. Sillion, and J. Dorsey, "Billboard clouds for extreme model simplification," *Acm Transactions on Graphics*, vol. 22, no. 3, pp. 689–696, 2003.
- [19] F. Tecchia and Y. Chrysanthou, "Real-Time Rendering of Densely Populated Urban Environments," in *Proceedings of the Eurographics Workshop on Rendering Techniques*, pp. 83–88, Springer Vienna, 2000.
- [20] M. Levoy and T. Whitted, *The Use of Points as a Display Primitive*, University of North Carolina, Department of Computer Science, 1985.
- [21] M. Zwicker, H. Pfister, and J. Baar V, "Surface splatting," *IEEE Transactions on Visualization Computer Graphics*, vol. 8, no. 3, pp. 223–238, 2001.
- [22] L. Kobbelt and M. Botsch, "A survey of point-based techniques in computer graphics," *Computers and Graphics*, vol. 28, no. 6, pp. 801–814, 2004.
- [23] R. Samuel and J. Buss, "Hardware-Accelerated Point Generation and Rendering of Point-Based Impostors," *Journal of Graphics Gpu & Game Tools*, vol. 10, no. 3, pp. 37–49, 2005.

