www.cadanda.com Computer-Aided Design AND APPLICATIONS

Using Virtual Reality Technology to Construct Computer-Aided Animation Material Development

Hao Luo[1] and Yue Wu[2]

[1]Department of Air, Sanjiang University, Nanjing, Jiangsu 210000, China, luohao_sju@126.com
[2]Department of Air, Sanjiang University, Nanjing, Jiangsu 210000, China, wuyue_sju@126.com

Corresponding author: Hao Luo, luohao_sju@126.com

Abstract. Computer-assisted virtual reality animation has created a new form of animation with its subversive visual experience. This paper proposes a dynamic mapping and load balancing strategy between multi-granularity tasks and heterogeneous parallel devices. A parallel task based on OpenCL for heterogeneous systems of CPU and GPU-Disturbing Particle Animation System is designed. We port existing CUDA-based parallel algorithms to OpenCL, thus breaking through the hardware limitations, and improving the portability and versatility of parallel algorithms. Combining the particle system animation material and the skeletal animation above, we design the principle of multi-tasking and heterogeneous parallel device mapping. According to the different OpenCL parallel acceleration capabilities of the CPU and GPU for different tasks, parallel tasks are allocated according to the acceleration ratio coefficient, and the performance is improved by reducing the waiting time of the device. This paper determines the relative optimal parameters of the random forest model through experiments, and through a comparison experiment with the decision tree model, shows the performance advantages of the random forest model that we designed and implemented for the automatic generation of animation. We counted and tested 614 animation materials, and finally rendered 379 animation materials. Among them, the animation active learning system was called when processing 369 animation materials, which showed the feasibility of the animation active learning system. The probability that the user likes the scene to be selected is improved, and the probability that the user does not like the scene to be selected is reduced. This shows the effectiveness of the computer-aided virtual reality animation system.

Keywords: Animation material; computer-aided; virtual reality; multi-task heterogeneous collaborative development
DOI: https://doi.org/10.14733/cadaps.2022.S5.155-166

1 INTRODUCTION

Compared with the past, the latest research in the field of virtual reality technology has a certain degree of progress, more extensive research content and more mature products [1]. Especially the application in the aerospace and other fields that developed earlier is relatively complete. From aircraft design that uses the combination of existing industrial drawing software and virtual reality technology to astronaut training exercises that simulate manned spaceflight, to the teaching and deduction of complex operations, the application of virtual reality technology in high-end technical fields is becoming more and more mature. At the same time, in the ordinary civilian consumer market, the output and quality of more advanced and richer virtual reality products such as games and animations have gradually increased. The characteristics of using virtual reality technology to create animation are mainly reflected in the richness of perception and the introduction of interactive modes [2]. Although traditional animation continues to progress and develop, from two-dimensional hand-painted to computer three-dimensional graphics to three-dimensional images, the three-dimensional sense and depth of field are gradually enriched. However, there is still an insurmountable gap between animation and the audience from the perspective of perception.

The audience of traditional animation is passive in accepting the selection of the screen, and there is no interactive feedback to the elements of the screen [3]. Just like in real life, we are locked with our eyes and hands and can only see the outside through a viewfinder. So this separation of perception creates a barrier to environmental immersion. The existence of virtual reality technology gives the audience the possibility to change the direction of the animation plot. With the development of animation production technology, its application fields have become more and more extensive, covering many fields such as animated feature films, animated advertisements, computer games and computer simulations. However, these animation productions have a shortcoming, that is, the production cycle is long, and the corresponding production cost is also high. The whole-process computer-aided animation automatic generation system incorporates the cutting-edge technology of artificial intelligence, which can greatly reduce the cost of animation production and shorten the animation production cycle. The character is the protagonist in the automatically generated animation, and the character is inseparable from the action. Therefore, the calculation of character action has become an important part of the automatic animation generation system [4]. In order to make the action more realistic and reasonable, the collision detection link is essential in the action calculation process. The research on collision detection has a long history and has a wide range of applications. It has become an important issue in the fields of computer graphics, computational geometry, simulation modeling, and robot path planning. In the animation automatic generation system, the motion calculation part will inevitably use this technology.

When the particle system is combined with skeletal animation and more modules, the research goal evolves from a single module to a multi-module multi-task system. For a single object node, in order to find the optimal parallel solution, it is only necessary to traverse various solutions on the CPU and GPU to find the optimal solution. For a system, it often contains multiple object nodes of the same type of node at the same time, and multiple types coexist. A VR system often contains multiple skeletal animation instances and multiple particle system instances at the same time, and the node complexity of the same skeletal animation may also be different. At this time, seeking the optimal solution of the parallel solution is no longer simply seeking the optimal solution of a certain type of node, but comprehensively considering all the target nodes, configuring some nodes on the CPU, and configuring another part on the GPU. It achieves CPU and GPU load balancing, which optimizes the overall parallel computing performance of the system. This article explores a solution for seeking optimal system performance for multi-node parallel computing of the same granularity and different granularities. According to the acceleration ratio of OpenCL on the CPU and GPU, it assigns the same ratio of task nodes to the CPU and GPU. This article evaluates the overall performance of the animation active learning system. First, it introduces the test environment and process of the experiment, and then counts the tested 614 animation materials. Finally, there are 379 animation materials that are rendered. Among them, 369 animation materials are called the animation active learning system shows the feasibility of the animation active learning system. Through the animation active learning system, the probability that the user likes the scene to be selected is improved, and at the same time, the probability that the user does not like the scene is selected, which shows the effectiveness of the animation active learning system.

2 RELATED WORK

In recent years, with the rapid development of computer animation technology, the application field of computer animation has been expanding day by day, and the social and economic benefits brought by it have also been increasing. At this stage, computer animation is mainly used in the film industry, TV titles and advertisements, scientific computing and industrial design, simulation, education and entertainment, as well as virtual display and 3D web. Fuchs et al. [5] pointed out that in many research directions of 3D computer animation, artificial intelligence-based animation research uses artificial intelligence technology for computer animation, thereby improving the automation and intelligence of animation production.

Lorenz et al. [6] believe that the characteristics of virtual reality are very obvious. It is an art form developed in today's highly developed electronic computer. This art form represents the development level of modern technology and is a new art form updated by modern media. In the form of expression, it is more comprehensive, interactive, and immersive, which is the most basic artistic characteristic of virtual reality art. Flury et al. [7] pointed out that the art of virtual reality is not just a simple display art, it focuses more on communication with the audience. At the beginning stage, let the audience get the joy of form in the process of experience (virtual reality is novel in form, and people will participate in and experience these new artistic creations because of curiosity, which is all pleased by the novel appearance). In the process, the audience is gradually infected by the artistic connotation given by the artist, and the audience's feeling is sublimated from the outside to the inside. This kind of performance is very different from the traditional art form. To measure whether virtual reality art is reasonable, it mainly depends on whether the form of expression adopted by virtual reality can meet the concepts and ideas that the artist wants to express, whether it can "please" the audience and whether the audience can accurately understand the meaning of the artist. Whether you can communicate with artists emotionally, these are the standard conditions for measuring virtual reality.

In virtual reality, all spaces are simulated, but it is real. Haider et al. [8] believe that the spatial impact of virtual reality on animation is not only visual and physical, but also manifested in psychological aesthetic space and perceptual space. The traditional way of expressing animation space is mainly focused on the simulation of three-dimensional things in a two-dimensional space, including color, light and shadow, sound, lens, motion, etc. for simulation. Although these seem to make the audience feel that the three-dimensional space really exists on the surface, the audience encounters difficulties when they want to explore further. The audience can only touch the side that the director shows to the audience. In other directions, the content of other perspectives does not seem to be the same as that of the director. The relationship shuts out the audience even more. However, virtual reality technology completely solves this problem. Zhang et al. [9] pointed out that virtual reality technology can allow us to have closer communication and contact with the displayed objects.

In the development process of the new system, not only the latest scientific research results are incorporated, but also various software is used as auxiliary. For example, Maya animation design software, which is the most popular in the international animation film industry, is used as the underlying animation software. MotionCapture 3D motion capture system and MotionBuilder professional 3D animation production software collects and process motion data. No matter it is the change of the system structure, the improvement of the algorithm or the addition of auxiliary software, the new system has a more powerful intelligent computing ability and graphic expressive power. As an important part of animation, motion calculation does not account for a lot in the ancient building wooden frame structure construction system. The main reason is that the object of motion calculation in the ancient building construction system is mainly simple structural components, and these are built. The action of the component is relatively simple. In order to make the animation automatic generation system more complete, the structure more reasonable, and the action calculation results more realistic, Hong and Brooks [10] combined with the experience of ancient architecture to build the system, and further adjusted the structure and algorithm, and added some functional modules. Detection is one of the added functional modules. This module provides corresponding solutions for various situations, and will be called many times during the action calculation process.

3 MULTI-TASK HETEROGENEOUS COLLABORATIVE PARALLEL COMPUTING OF VIRTUAL REALITY SYSTEM FOR ANIMATION MATERIAL DEVELOPMENT

3.1 OpenMP Multi-Threaded Task-Level Parallelism

When a single thread executes on a single CPU physical core, it will approach the theoretical maximum performance. Each CPU often contains 4 or more core processor units. In theory, the development of multi-threaded parallel programs will achieve as many speedups as the number of cores. For example, a processor that contains 4 physical cores and supports 4 physical threads will get a speedup of 4, and a processor that contains 4 physical cores and supports 8 threads will get a speedup of 8.

OpenMP is an open multithreading method library. It has good code portability and performance portability between different CPU hardware manufacturer models, different compiler software, and different operating systems. OpenMP has been built into the most commonly used compilers such as VC for Windows, GCC for Linux and Mac. OpenMP can quickly switch between multi-threaded and single-threaded states, and programmers only need to write very little code. For situations where the algorithm structure is not very complicated and the algorithm has a huge amount of calculation, OpenMP is particularly suitable for selection. In addition, you can also use other multi-threaded libraries, such as Intel TBB (Threading Building Block), to meet more complex and diverse multi-threaded programming needs. This article uses VC language to call OpenMP to realize the multi-threaded transformation of the algorithm under the Windows 7 operating system. The transformed multi-threaded algorithm can be easily transplanted and reused.

No additional libraries are needed to call OpenMP. You add the compilation option "/openmp" when compiling, or enable the "OpenMP Support" setting in the VC project properties, which is located on the C/C++ Language page. Then you import "omp.h" in the header of the file that needs to execute multithreading, and then you can call the API methods provided by OpenMP and the #pragma omp series of macro instructions. These methods include modifying the number of threads, obtaining thread IDs. One of the most commonly used occasions in OpenMP is to execute loops in parallel. The loop body is split into multiple threads to execute in parallel. To achieve this multithreading, you only need to call a #pragma omp instruction, that is, call "#pragma omp on the line before the for statement. parallel for ".

Using OpenMP is very simple, but not all programs can be parallelized, and some programs cannot achieve performance improvement even if they are parallelized. If you want to take advantage of the parallel acceleration capabilities of multithreading, the algorithm should meet the following characteristics. First of all, there is little or no communication between threads to avoid thread dependence, and one thread waits for the running result of another thread; secondly, there is little or no sharing of data between threads to avoid resource competition, which causes threads to be in a waiting state and consumes time; third, the data access addresses of adjacent threads should be consecutive or similar to give full play to the pre-reading and reuse efficiency of cached data.

3.2 Particle System Simulation

The primary task of the simulation of special effects in virtual scenes is to establish the model of special effects, that is, the static or dynamic pictures presented in the computer such as rain and snow in the real environment, animation materials, etc. This article adopts the particle system modeling method. The basic idea of the particle system is to use a large number of tiny particle primitives with certain life and attributes as the basic elements to represent irregular and fuzzy objects. The "generation", "activity" and "death" of particles enable the particle system to generate a series of moving scenes.

Particle system theory is one of the most mature theories used to describe irregular objects so far. It uses a set of methods that are completely different from the previous modeling and drawing systems to construct and draw scenery. The scenery is defined as tens of thousands of irregularities. It is composed of particles that are randomly distributed, and they change shape and move constantly. Therefore, particle systems are widely used in the simulation of natural scenery. Structured particle systems are usually used to simulate trees, leaves, water droplets, grass, rainbows and clouds, etc.; random particle systems are usually used to simulate fireworks, explosions and snow, etc.; directional particle systems are usually used to simulate deformable objects or rigid objects. The computer-aided virtual reality animation material development particle system is shown in Figure 1.

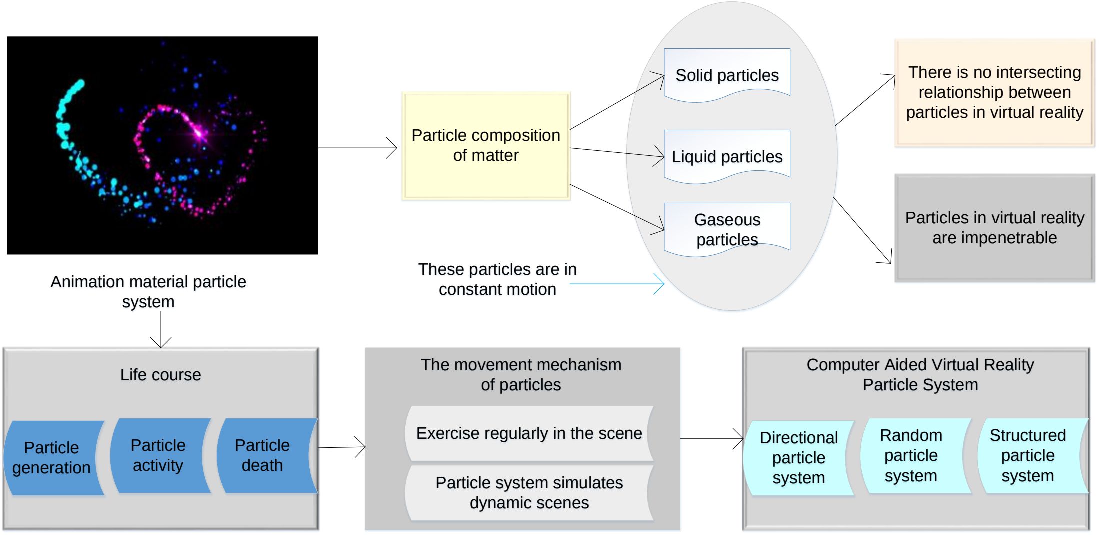

Figure 1: Computer-aided virtual reality animation material development particle system.

After the initialization of the new particles is completed, the particles start to move in the system environment according to the speed and direction at the time of initialization, and the other attribute values of the particles also change over time. The spatial position of the particle is mainly determined by the previous position of the particle, the velocity of the particle and the time interval of each frame. Assuming that the time interval between the j+1 frame and the j frame is $\triangle t$ · , the spatial position of the particle at the j+1 frame is:

$$PositionX_j = PositionX_{j-1} - 0.5\Delta t \bullet VelocityX_{j-1} \quad (1)$$

$$PositionY_j = PositionY_{j-1} - 0.5\Delta t \bullet VelocityY_{j-1} \quad (2)$$

$$PositionZ_j = PositionZ_{j-1} - 0.5\Delta t \bullet VelocityZ_{j-1} \quad (3)$$

The velocity of a particle is determined by the initial velocity of the particle and the external force experienced by the particle. Common forces in the particle system include gravity, wind, buoyancy, and the force generated by collisions between particles. In order to simulate the movement of particles in a complex environment. By controlling the parameters of these fields, the trajectory of particles can be controlled to make the particles move according to a certain law. In this paper, Perlin Noise is used to simulate wind field interference sources. Assuming that the acceleration generated by the particle and the external force F is Acceleration, then the velocity of the particle at frame j+1 is:

$$VelocityY_j = VelocityY_{j-1} - \Delta t \bullet Acceleration \quad (4)$$

Among them, Velocity and Acceleration are vectors in a three-bit space. The change of other properties of particles is generally calculated by the following formula:

$$\Pr opertyY_j = \left|\Pr opertyY_{j-1} - \Pr operty\Delta t\right| \quad (5)$$

This paper uses OpenCL-based parallel computing to improve the original particle system algorithm, mainly to improve the coordinate movement and attribute update of particles. First, according to the concept of particle system, we establish the data structure and algorithm for the particle system. The coordinate reset update and animation rendering algorithm flow is shown in Figure 2.

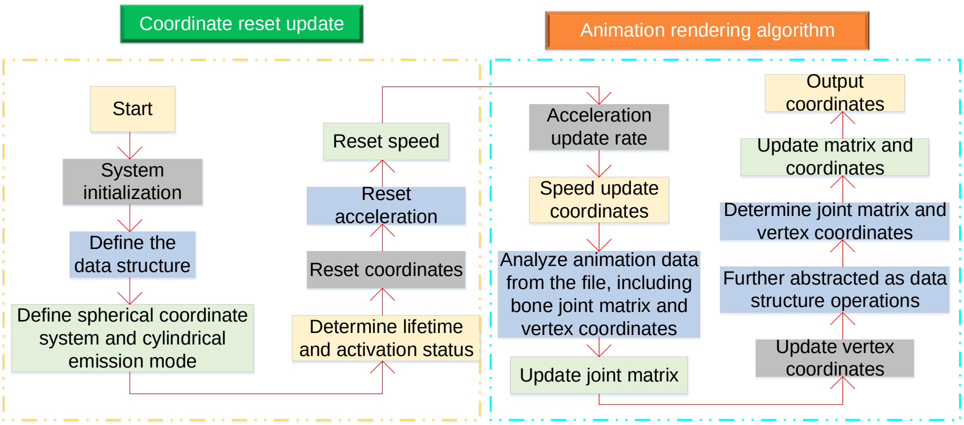

Figure 2: Coordinate reset update and animation rendering algorithm flow.

3.3 CPU and GPU Heterogeneous Collaborative Processing VR System Multi-Task Animation Material Development

The main research goal of the above content in this article is a single object node. For a single object node, in order to find the optimal parallel solution, it is only necessary to traverse the various solutions on the CPU and GPU to find the optimal solution. For a system, it often contains multiple object nodes of the same type of node at the same time, and multiple types coexist. Take the skeletal animation and particle system involved in this article as examples. A VR system often contains multiple skeletal animation instances and multiple particle system instances at the same time, and the node complexity of the same skeletal animation may also be different. This article will also call the node complexity for the granularity. At this time, seeking the optimal solution of the parallel scheme is no longer simply seeking the optimal solution of a certain type of node, but comprehensively considering all target nodes, configuring some nodes on the CPU, and configuring another part on the GPU. According to different types and granularities, this article will discuss multi-task parallelism in the following four situations.

After parallel computing based on OpenCL, the speedup ratio on the CPU is 3.9 and the speedup ratio on the GPU is 5.0. The ratio between the two is 0.44 to 0.56. If there are multiple skeletal animations, they are allocated to the CPU and GPU at a ratio of 44% to 56%. Assuming there are N nodes, the numbers of CPU and GPU allocation nodes are N*0.44 and N*0.56, respectively, and the numbers are rounded up. This paper uses 2, 4, 8, 16 nodes as examples to conduct experiments, and found that when the ratio of the number of nodes allocated to the CPU and GPU is 0.44 to 0.56, the overall performance is optimal. Similarly, for the particle system, after performing parallel calculations based on OpenCL, the number of particles is set to 10,000, the acceleration ratio on the CPU is 1.9, and the acceleration ratio on the GPU is 13. The ratio between them is 0.13 to 0.87. If there are multiple particle systems, they are allocated to the CPU and GPU at a ratio of 13% to 87%. Assuming there are N nodes, the numbers of CPU and GPU allocation nodes are N*0.13 and N*0.87 respectively, and the numbers are rounded up. This paper uses 10 nodes as an example to conduct experiments, and it is found that when the ratio of the number of nodes allocated by CPU and GPU is 0.13 to 0.87, that is, 1 to 9, the overall performance is optimal.

4 EXPERIMENTAL SIMULATION AND ANALYSIS

4.1 Experimental Environment and Process

The computer-assisted animation material automatic generation system is the carrier for the realization of the animation active learning system. The hardware equipment of the animation generation system includes a CPU (model Intel E5430), the system's master control server (operating system is 64-bit Windows XP HP graphics workstations), 13 rendering servers (the rendering server is Inter(R) Xeon(R) CPU E5410 @2.33GHZ 2.33GHZ Memory 2GB Sugon blade unit). The system software environment mainly uses 32/64-bit Windows operating system, ontology development tool Protégé, database SqlServer2005, Java development tool Eclipse, C++ development tool Microsoft Visual Studio 2013, animation production software Maya2009 and rendering tool BackBurner rendering software.

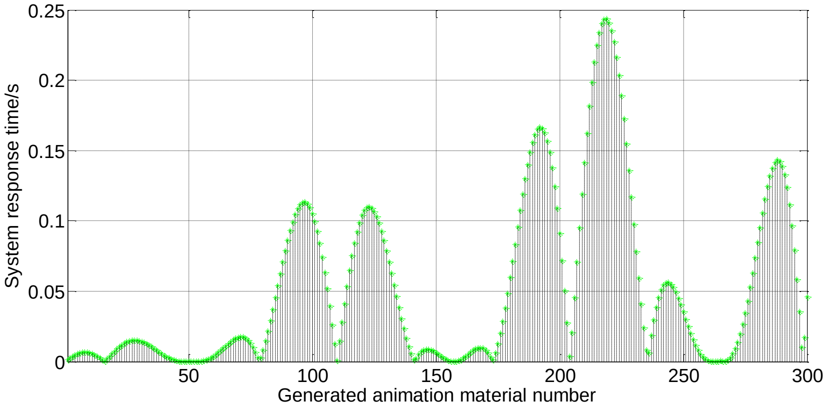

Figure 3: Real-time test of computer-aided animation material automatic generation system.

The qualitative planning module of the computer-aided animation material automatic generation system is based on Java language, and the quantitative calculation module is based on C++. The interactive module and decision module of the animation active learning system are implemented based on Java language, and the training module is implemented based on C++ language. The animation active learning system is a part of the computer-assisted animation material automatic generation system, and its testing process is completed by the computer-assisted animation material automatic generation system. The animation material received by the animation generation system can be completed by sending the animation material through the website or mobile phone. The real-time test of the computer-aided animation material automatic generation system is shown in Figure 3.

The animation generation system can view historical animation material animations on a specified date or within a specified time period. In addition, the generation system can view the key system data information of each animation material, such as information extraction IE files, plot qualitative planning ADL files, and candidate scene files and quantitative planning files can be selected. By viewing these files, you can learn the detailed generation process of each animation material to verify the program operation effect and provide data analysis support for further research in the future. Checking the effect of the learning system is mainly done by viewing the plot qualitative planning ADL file, and using candidate scene files. The distribution of the key information acquisition quantity of the computer-aided animation material automatic generation system is shown in Figure 4.

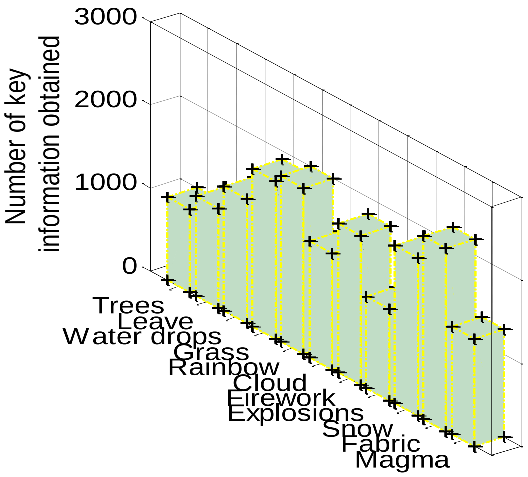

Figure 4: The distribution of the key information acquisition quantity of the computer-aided animation material automatic generation system.

4.2 Feasibility Experiment of Animation Active Learning System

There are 379 animation materials rendered, of which 369 animation materials are called by the animation active learning system, which verifies the feasibility of the animation active learning system through a large number of animation materials. In this section, an animation material will be used to verify the feasibility of the learning system.

In the plot planning part, according to the theme, the template selects 4 related candidate scenes as shown in Figure 5. Then the 4 candidate scenes are scored in turn according to the scoring items, and the user preference value userfavorvalue returned by the learning system is used as one of the scoring items. Figure 6 shows the scores of the 4 scenes in the animation material. A scene with a higher score is more likely to be selected as the final use scene. The animation material with the highest score is selected as the rendering scene. After determining the scene, the animation generation system plans the model, action, layout, etc., each part will have corresponding planning information, and the planning information of the plot planning part is stored in the ADL file.

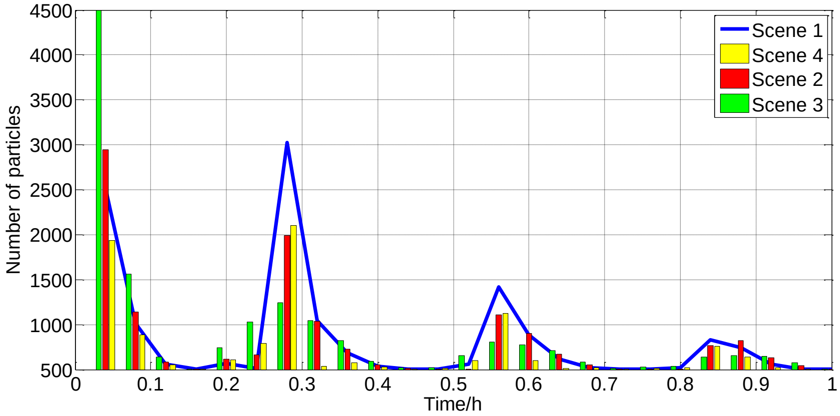

Figure 5: Particle distribution of 4 candidate scenes of animation material.

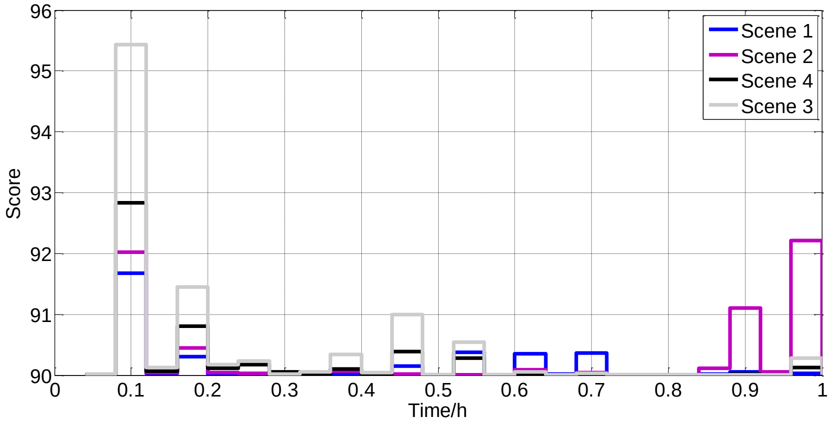

Figure 6: The ratings of the 4 scenes in the animation material.

4.3 Effectiveness Experiment of Animation Active Learning System

An animation scene with a user rating of 90 points under the text content of a specific animation material is recorded as the user's favorite scene. The background scene of the animation material animation hurry.ma is the user's favorite scene, which means that the user thinks that the hurry.ma scene is a perfect display of the content of the animation material text. From the user’s feedback on the animation, we find out 10 animation materials with a score of 60, and extract the corresponding animation material text content and selected scenes, and 10 animation materials with a score of 90 to extract the corresponding animation material text content. The animation material contents of these 20 animation materials are not related to each other, and these 20 animation materials are used for comparative experiments.

The specific steps are: (1) Input these 20 animation materials one by one into the computer-aided animation material automatic generation system without animation active learning system to obtain the scene selected by the system. The same animation material is repeatedly tested 30 times, and the user is calculated Like/dislike the scene is selected probability. (2) Input these 20 animation materials one by one into the computer-assisted animation material automatic generation system under the guidance of the animation active learning system to obtain the scene selected by the system. The animation materials with the same content are repeatedly tested 30 times, and the user likes/does it Probability of favorite scene being selected.

The animation active learning system can increase the selection probability of the user's favorite scene corresponding to the specific animation material text, and reduce the selection probability of the user dislike the scene, but it does not achieve full coverage. The difference in the increase/decrease range is mainly due to the different animation material text. The selected candidate scenes are different, which shows that the animation active learning system can learn the user's preferences for the scene, so as to help the computer-aided animation material automatic generation system to make a more satisfactory choice for the user. Figure 7 shows the comparison of the selection probability of animation material scenes with a user score of 60 points. Figure 8 shows the selection probability comparison of animation material scenes with a user score of 90 points.

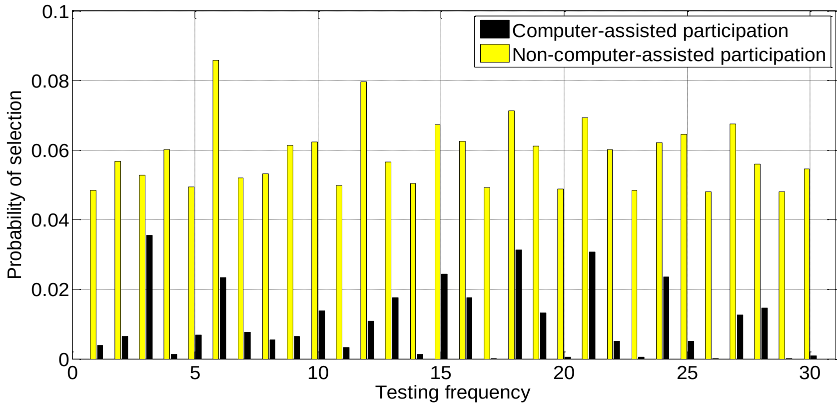

Figure 7: Probability of selection of animation material scene with a user score of 60.

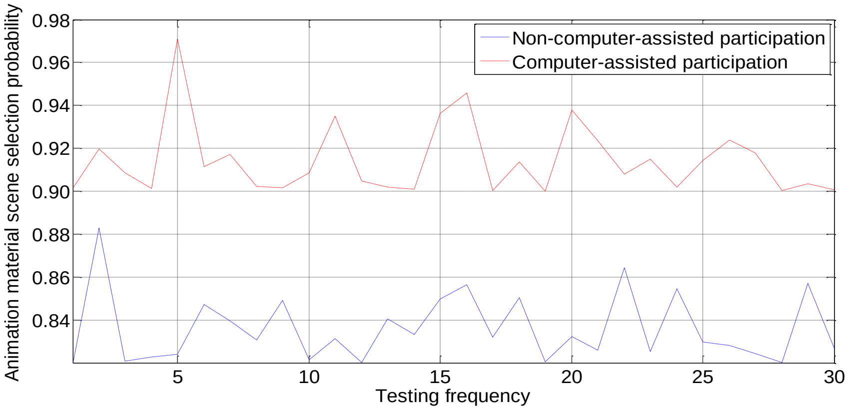

Figure 8: Probability of selection of animation material scene with a user score of 90.

5 CONCLUSION

This article improves the portability of the GPU animation matrix palette algorithm. The original CUDA-based algorithm relied on a specific GPU, and the GPU-oriented matrix palette algorithm based on OpenCL is generally applicable to various GPUs. Taking the CPU serial algorithm and the traditional parallel algorithm based on SSE and OpenMP as the performance reference benchmarks, the speedup ratios are 3.9 and 1.5 respectively. This paper designs a parallel optimization scheme for the animation matrix palette algorithm, and designs an algorithm to automatically select the optimal scheme. Including the latest is OpenCL, which can be used for both central processing unit CPU and graphics processing unit GPU. In addition, special parallel schemes are designed for CPU and GPU. Among them, OpenMP and SS are specifically for CPU; GLSL and CUDA are specifically for GPU. On all different CPU and GPU configurations, it automatically finds a feasible and optimal solution. This paper determines the relative optimal parameters of the random forest model through experiments, and through a comparison experiment with the decision tree model, shows the performance advantages of the random forest model that we have achieved for the automatic generation of animation. At the same time, this paper designs experiments to verify the feasibility and effectiveness of the animation active learning system. This article is the first time for the computer-aided animation material automatic generation system to design and implement an "active" animation learning system. After the animation generation system receives the animation material text, it selects the background scene that is more popular by the user, and improves the user of the animation product. The animation active learning system is closely connected with all aspects of the animation generation system, and the effect of the system is closely related to these links. Further research work includes the expansion of learning modes, the exploration of other algorithms, the optimization of system parameters, and the reduction of user and system administrator participation.

Hao Luo, https://orcid.org/0000-0001-7146-9736
Yue Wu, https://orcid.org/0000-0002-4279-3112

REFERENCES

[1] Chang, W.: Virtual Reality Filmmaking Methodology (Animation Producing), Techart Journal of Arts & Imaging Science, 3(3), 2016, 23. https://doi.org/10.15323/techart.2016.08.3.3.23

[2] Gan, B.; Zhang, C.; Chen, Y.; Chen, Y.-C.: Research on Role Modeling and Behavior Control of Virtual Reality Animation Interactive System in Internet of Things, Journal of Real-Time Image Processing, 18, 2021, 1069-1083. https://doi.org/10.1007/s11554-020-01046-y

[3] Henshall, G.-I.; Teahan, W.-J.; Ceny, L.-A.: Virtual Reality's Effect on Parameter Optimization for Crowd-sourced Procedural Animation, Visual Computer, 34(9), 2018, 1255-1268. https://doi.org/10.1007/s00371-018-1501-2

[4] Kang, N.; Bai, J.; Pan, J.; Qin, H.: Interactive Animation Generation of Virtual Characters Using Single Rgb-d Camera, The Visual Computer, 35, 2019, 849-860. https://doi.org/10.1007/s00371-019-01678-7

[5] Fuchs, C.; Allmacher, C.; Klimant, P.; Putz, M.: Holistic Animation and Functional Solution for Digital Manikins in Context of Virtual Assembly Simulations, Procedia CIRP, 97, 2021, 15-20. https://doi.org/10.1016/j.procir.2020.05.198

[6] Lorenz, M.; Spranger, M.; Riedel, T.; Pürzel, F.; Wittstock, V.; Klimant, P.: CAD to VR – A Methodology for the Automated Conversion of Kinematic CAD Models to Virtual Reality, Procedia Cirp, 41, 2016, 358-363. https://doi.org/10.1016/j.procir.2015.12.115

[7] Li, C.; Zuo, Z.-X.; Huang, X.-Y.: Stereoscopy Animation Actualizing under Practical Virtual Reality, Computer Simulation, 11(6), 2002, 84-86. https://doi.org/10.3969/j.issn.1006-9348.2002.06.027

[8] Haider, A.-M.; Abdulrahman, A.-A.; Ali, A.; Rashid, M.: A Systematic Approach to Parameter Selection for Cad-virtual Reality Data Translation Using Response Surface Methodology and Moga-ii, Plos One, 13(5), 2018, 1-19. https://doi.org/10.1371/journal.pone.0197673

[9] Zhang, F.; Wei, Q.; Xu, L.: A Fast Simulation Tool for Fluid Animation in VR Application Based on Gpus, Multimedia Tools and Applications, 79(23), 2020, 16683-16706. https://doi.org/10.1007/s11042-019-08002-4

[10] Hong, X.; Brooks, S.: 3d Object Clouds: Viewing Virtual Objects in Interactive Clouds on Mobile Devices, IEEE Transactions on Visualization and Computer Graphics, 26(3), 2020, 1442-1453. https://doi.org/10.1109/TVCG.2018.2873724