

DESIGN AND IMPLEMENTATION OF GAME SIMULATION BASED ON PARTICLE PROPERTIES

Yuqiang Sun, Yixuan Xing, Yuanyuan Li, Yong Lu, YuwanGu*

School of Information Science & Engineering, ChangZhou University, Jiangsu, Changzhou, 213164, China

Abstract :- This paper introduces a calculation method for the simulation of smoke fluid based on particle attributes, the complex and abstract wind and dust and other objects in nature is described of adding and modifying through the controllable particle swarm properties. In the computer scheduling algorithm, because the data volume is huge, although the most functions and attribute transformation has been solved by the method from the library of the software itself. But many places still to write out some short code, so as to simplify the complexity of calculation and the model becomes more practical.

Keywords - complex system; Particle properties; the coordination degree

I. INTRODUCTION

With the improvement of living standards, the proportion of spiritual consumption in consumption occupies is also growing. On behalf of the contemporary spirit of consumer goods is the game itself. Today, in people's recognition more and more high to the game, plane game through the accumulation of ordinary pixel has been slowly out of the time. It is not only the pursuit of operating pleasure when playing a game, but also the game screen has become an important standard to measure a game is good or bad now[1-2]. With the development of technology of computer graphics, the game world by two-dimensional original into a 3D world today. The three-dimensional world is an emphasis on authenticity and beauty picture; this will increase the player's sense and game fun[3]. Major game companies have invested a lot of effort in developing the game scene.

In this research, the traditional polygon model simulation is very quickly used by the industry, but simulation of the complex substances (such as wind, fluid, smoke and so on) in the nature is always a difficult problem. Due to no fixed shape and variability of state of these substances, it is not simulated by a systematic algorithm. And the study of particle swarm get rid of constraints based on the rough polygon, add to the complexity of specific texture images, the game screen production level become more and more high, and some have reached the level of production of the film[4-5].

Therefore, research on particle properties is trend of direction of game simulation[6]. This paper introduces a calculation method for the simulation of smoke fluid based on particle attributes, the complex and abstract wind and dust and other objects in nature is described of adding and modifying through the controllable particle swarm properties. And design the attribute definition and conversion functions which the software library itself does not have, in order to simplify the computation complexity, the model becomes more practical. The design of attribute definition, changes and conversion function of simulation of the tornado of

particle group is as the main line, discusses the call and connection between the model and the library.

II. THE ESTABLISHMENT OF MEASUREMENT MODEL AND THE DRIVE PROPERTIES

For the ground rotating flow of tornado, it needs to build a grid as the basic chassis[7]. Call the scatter node after establishing grid; grid will be converted into point set which arranged in the scope of a grid plane. At this time point is the average arrangement, in order to display the characteristics of sparse dense of tornado, then call the ramp node in the chop, create a rounded black and white gradient map. Gradient map in chop cannot be called directly in VOP, so we need to add the vopsop calculator; in the vopsop using the colormap function to obtain address of the gradient map in the chop and then color (Cd) property is passed to the grid. Finally the color attribute is passed to the position attribute in the attribute parameter of scatter node; you can get a sparse, dense point group[8-9].

Normal attribute of point is as driver attribute of point after the definition of point position, normal vector of (0,1,0) is created when the points are created using the scatter node. First normal vector need to rotate along the center point, due to the existence of a normal vector of (0,1,0), as long as the position vector of each point is as another vector, vector which is calculated using the cross algorithm in the vopsop calculator (the vertical vector of two normal vector) is restricted using fit function, so that its size in 0-1, add the calculated vector together the position vector through constraint of fit function. You can get the method of vector group which is like a whole whirlpool. Now create a new vopsop calculator, the normal vector of (0,1,0) will be multiplied by the value of red channel in the color attribute (get through \$CR), get a gradient effect from quantitative change to center increase of Y axis, finally add the previous result. Finally, get the normal group which is the spiral advance to the center of spiral in the XZ plane, increased to center point in Y axial.

Publication History

Manuscript Received : 30 November 2015
Manuscript Accepted : 16 December 2015
Revision Received : 25 December 2015
Manuscript Published : 31 December 2015

For the air flow of tornado, a block of 50 straight lines are created using line nodes and resample nodes. Continue to call the point node, use the expression $\$TX + \sin(\$PT * \text{ch}(\text{"freq"})) * \$TY * \text{ch}(\text{"amp"})$ in X, including "freq", "amp" is custom expression in the node, $\$TZ + \sin(\$PT * \text{ch}(\text{"freq"})) * \$TY * \text{ch}(\text{"amp"})$ in z. After defining the shape of the spiral and then copying a point node is used for making the tornado body swing. Here only to reduce "amp" (amplitude) parameters. When the two curve shape made, enter the vopsop calculator to add directly each point position of two lines, this data type conversion, the vector type converted into float type, and then converted back to vector type, role is to modify the value of point of Y axis is 0 then it do not participate in the calculation. Next, add the normal attribute to each point used to brake in dop. Using the point $(\text{".."} + \text{opinput}(\text{"."}, 0), \$PT + 1, \text{"P"}, 0) - \TX to define the direction of the X axis, $\text{point}(\text{".."} + \text{opinput}(\text{"."}, 0), \$PT + 1, \text{"P"}, 1) - \TY to define the direction of the Y axis, $\text{point}(\text{".."} + \text{opinput}(\text{"."}, 0), \$PT + 1, \text{"P"}, 2) - \TZ to define the direction of the Z axis. Finally, a total of normal vector is composed. This calculation is that vector difference of point of the ID N and ID n+1 is as the method of vector n. Call the polywire node, spiral line can be converted into a volume model. Using a very important point of offset node in volume mode, the role of the node is that an offset surface is created in the original surface. This operation establishes an implicit function input geometries. Then original surface is shifted on fixed offset using the implicit function. So the relationship between the point and surface is described, when the point on the surface is obtained the value of 1, whereas is obtained the value of 0. Points on the surface obtain a 0 - 1 value according to the location. This time all points of air flow is created using a scatter node. These nodes have no normal attribute, so must pass property by attribute transfer node, the node will calculate the point of property transfer which is nearest the point of each attribute, then the property transfer. The advantage of using the node carries on transfer property even the different types of data. Add the vopsop calculator; add the noise function in a calculator, increase the randomness of particle positions, which makes the simulation more realistic.

After two groups of analog synthesis to a scene, a final shape of static tornado is generated using volume from attrb. The role of Volume from attrb node is passing the attribute of point to the volume of voxels. Fill $A' = A * m + A$ in the expression properties to transfer myvel (namely normal in order to facilitate the distinction of variables for a name). This time volume contains attributes of 0 (vel, x), 1 (vel, y), 2 (vel, z), A white mist gradient is created around each point through the attribute, concentration changes according to the value of the attribute in volume, 1 is white, 0 is transparent. Linear interpolation is used between 0 and 1. Finally, you can get a static tornado. The operation in SOP, chop end.

III. DOP LIBRARY CALLS

To complete the operation of sop, chop node, just give tornado a static appearance, really need long time settlement within the operation of DOP. DOP means dynamic operate & parameter (dynamic operators and parameters). Thus the DOP is used to control the animation, the most important thing to control the animation is using one engine, in the control of smoke movement, and the software provides the

pyro solver engine for us. The internal core of engine is actually an iterator; each frame state is determined by the front frame on the calculation. Pyrosolver node needs to use several object variables.

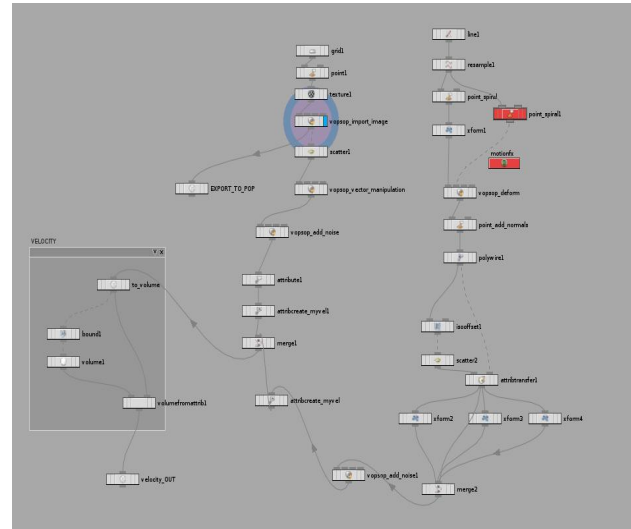


Fig. 1 SOP node graph

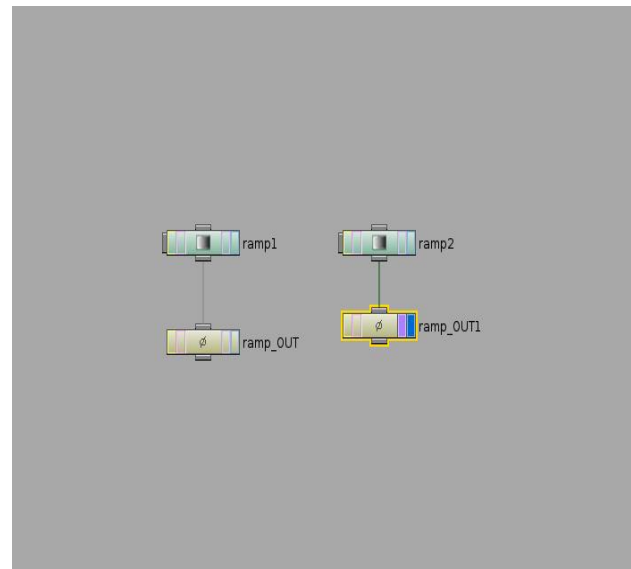


Fig. 2 Chop node graph

Firstly object needs a density with real representation to create the density of smoke body, and then the object needs a velocity vector to define velocity value of each voxel, calculation of buoyant smoke body using temperature property represent with a real number. In this simulation we first need to give pyrosolver a smoke object node, the role of the node is that the tornado created in vop is imported into dop, and also data is imported into dop. Only node path needs to be assigned to path parameter of the smoke object node. The pyrosolver needs another parameter which is given by gas resize fluid dynamic node, used to constrain the size of simulation space. Calculation of smoke fluid is the impact of voxel to space, it has very little influence for space when the distance between space with voxels is far, if the calculation of those nodes of distant position will greatly increase the time of fluid settlement, and the final effect is

almost confined with computational constraints space. This can greatly save the time of fluid settlement, meet the need of the work flow. Finally we will call volume created in vop to the dop, using the source volume node to directly call, property pages in volume will be call to the dop together. Finally, add the gravity node to the whole scene, add gravity to scene, and make the motion of tornado is fit to the laws of nature. So far, the operation in dop is completed.

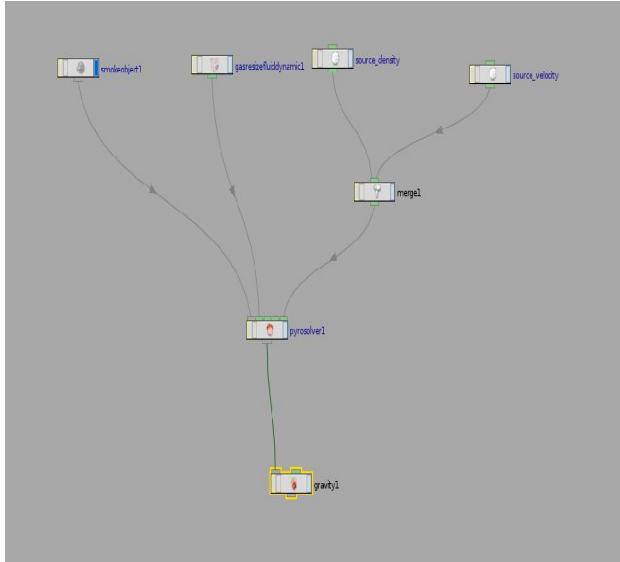


Fig. 3 Dop node graph

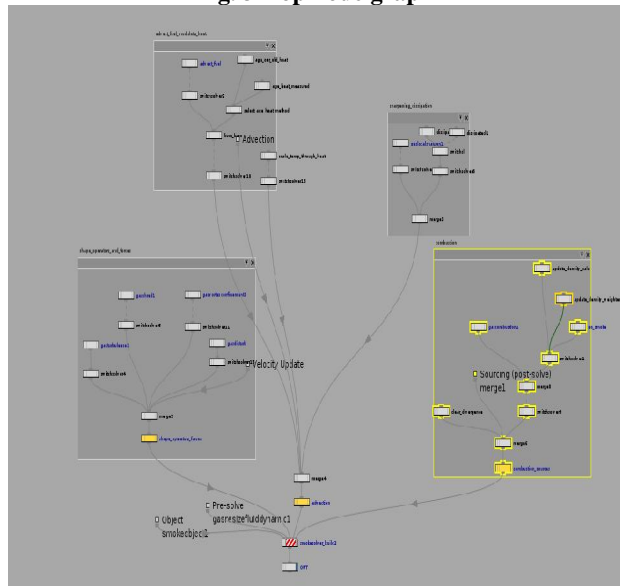


Fig. 4 Dop node graph

IV. DRIVEN PARTICLE NODES

Through the previous operation, it has completed the definition of tornadoes form and animation. Next you need to do dust, in front of the SOP, when doing the gradient map, it gives a null node. Thus, gradient map expressed by null node in sop is called directly using an object merge node in pop. Still add the scatter node to make surface into a point, continue to add the popnet node. The popnet node is associated with houdini particle, all particles nodes are completed in the popnet node. Create a good point set as the

emitter, custom emitter parameters in popnet. It has not to define too much particle motion attribute here, here the particles is tornado blowing dust, so the tornado is drive of these particles. In houdini, volume drive particles can through advectbyvolumes node, the node takes volume property to transmit to particle, to achieve particles drive. The benefit of this node is that can dynamically read attribute in volume, then can realize the real-time transformation. Any ID particle information can be read using the point function, if the reinspection you can see the point has get accelvel, P property in the volume. These three properties make point to dynamically link to volume above, mobile following volume. As if the tornado blew the dust.

After the pop operation, just created some of the most simple solid color layer in shop, the particle swarm gets for sand material.

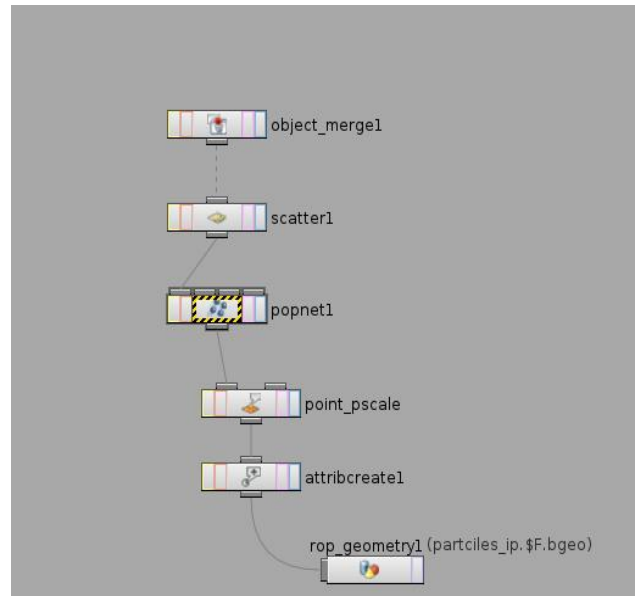


Fig. 5 Pop node graph

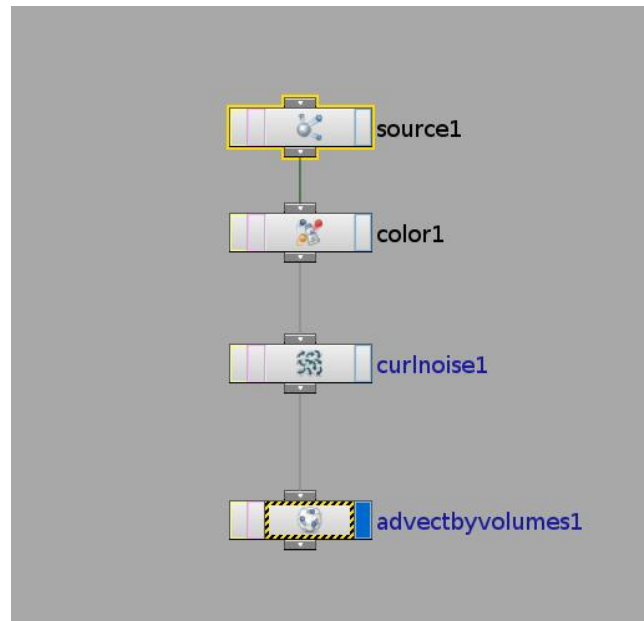


Fig. 6 Pop node graph

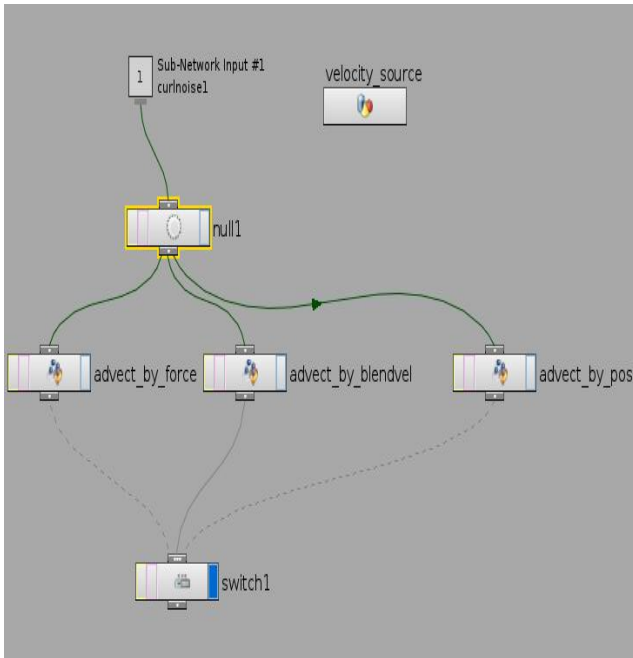


Fig. 7 Pop node graph

V. THE OPERATION MODULE CONNECTION

Houdini divided into several modules of OBJ SOP SHOP VOP POP DOP COP CHOP ROP, the reason why are all OP, actually it is the abbreviation of operate & parameter (operators and parameters). These modules are respectively OBJ (object transform layer) SOP (Geometry sharp layer) SHOP (shader material) VOP (graphical programming vex&rsl tools) POP (particle) DOP (dynamics) COP (post synthesis) CHOP (channel editor) ROP (render output). The default is on the object root path, alt+9 showed that tree list can view path structure.

These modules are completely independent, you can repeated call other modules through creating OPnetwork from any of module, there is no absolute relationship between father and son (Note: not to create the module itself, but to create a OPnetwork interface which can be independently call function module, press the Enter key to enter OPnetwork to create network structure of the module function) OPnetwork node of any module can be regarded as a folder. If no file on the folder, then the folder itself has any meaning, of course, you can copy the contents of the file or the whole folder to any the same type path. Tree list can put it as the system path and file list. If you often need to do compile work, you must know if the path is not set to PATH, it also can be compiled, but it must be put CD into the path of internal environment of the HOUDINI to use script, it also must put opcd /path/ into the path, we can correctly operate.

VI. DESIGN EXAMPLES: TORNADO EFFECT

1) Sop point set graph (normal attribute, Blue is the direction of vector)

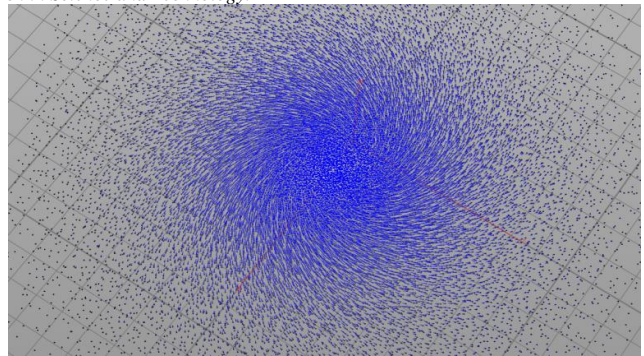


Fig. 8 Sop point set graph

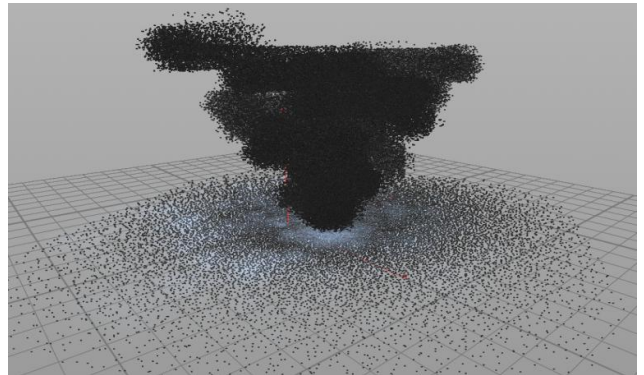


Fig. 9 Sop point set graph

2) Sop volumn graph

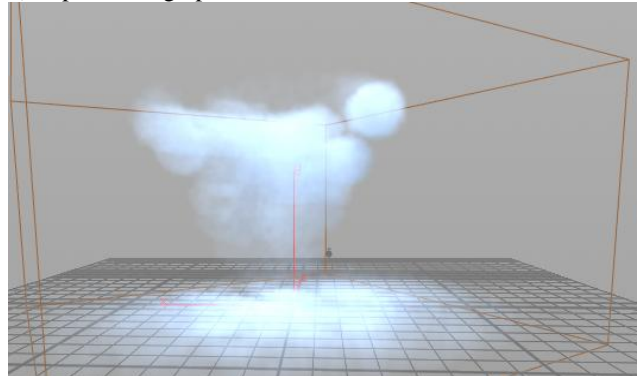


Fig. 10 Sop volumn graph

3) The effect graph of dust

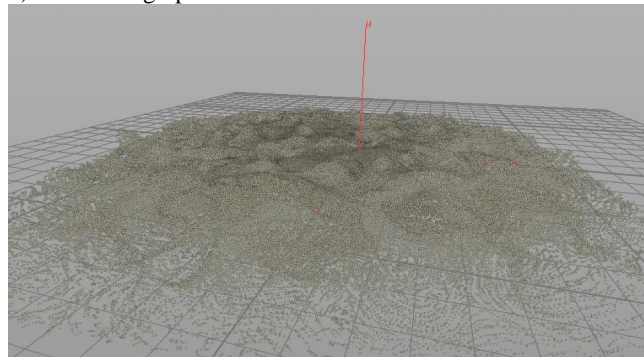


Fig. 11 The effect graph of dust

4) The last effect graph

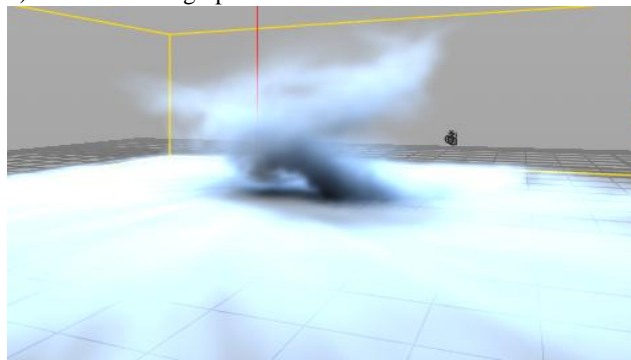


Fig. 12 The last effect graph

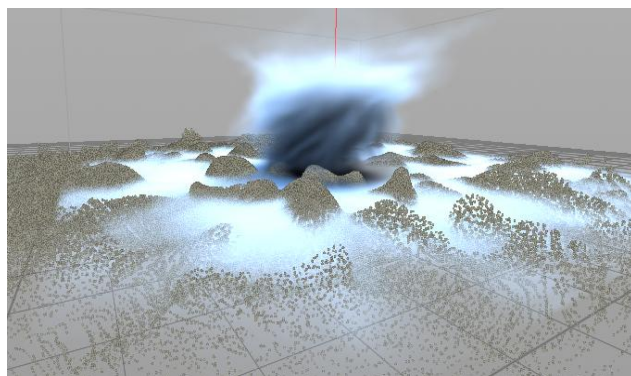


Fig. 13 The last effect graph

VII. CONCLUSIONS

Everything is composed of the most basic point, for the irregular objects are hard to describe, using the point set to simulate is the best method. The advantage of points is that they can move freely to form any wanted graphics, adding different properties to the point that can macroscopically define the form and animation of a component object [10]. But it is the most difficult to control a point set, because of its controllability is too flexible. When the numbers of points become more and more, and the status of simulation is more complex, the calculation time of these results is increasing geometrically; to obtain feedback time is longer, it produces energy loss for the whole control and the adjustment of details. This is the biggest problem when simulating a point. Therefore, this paper emphasizes on dealing with each parameter well, first to simulate with low accuracy, which can greatly save time, improve work efficiency.

ACKNOWLEDGMENT

We like to express sincere appreciation and deep gratitude to all participants in this work.

REFERENCES

- [1] Zhang Renjing, Liu Bin. Special technology of drawing and management of 3D game scene[J]. Journal of Guizhou Normal University (NATURAL SCIENCE EDITION). 2011, (01): 74-77, 87
- [2] Zhang Xiaowei. Design of 3D rendering engine based on OpenGL[J]. Software Guide. 2009, (08): 185-186
- [3] Joe. Study on Houdini algorithm[M]. Beijing: Machinery Industry Press. 2012

- [4] Foley J D, Dong Shihai. Introduction to computer graphics[M]. Beijing: Machinery Industry Press. 2005
- [5] Dietrich Kammer, Jan Wojdziak, Thomas Ebner, Ingmar S. Franke, Rainer Groh. A component-oriented framework for experimental computer graphics. computer Standards & Interfaces, V34(1). 2012, P:93-100
- [6] Jung W.Suh, Youngmin Kim. Example in Computer Graphics. Accelerating Matlab with GPUs. 2014. P:157-191
- [7] Hong Jeong-Mo, Kim Chang-Hun. Animation of bubbles in liquid[J]. COMPUTER GRAPHICS FORUM, 2003, (03): 253-262.
- [8] Liu Youquan, Liu Xuehui, Zhu Hongbin. Fluid simulation in computer animation based on physics[J]. Journal of Computer Aided Design & Computer Graphics. 2005, (12): 2581-2589
- [9] Alun Evans, Marco Romeo, Arash Bahremand, Javi Agenjo, Josep Blat. 3D graphics on the web: A survey. Computers & Graphics, V41, 2014, P:43-61
- [10] Niu Lianqiang, Feng Haiwen, Hou Chunguang. C language program design: The theory and application of Engineering[M]. Beijing: Publishing House of electronics industry. 2013