A Fast Scene Constructing Method for 3D Power Big Data Visualization

Zhao-Yang Qu and Jing-Yuan Huang
School of Information Engineering of Northeast Dianli University, Jilin 132012, China
Email: qzywww@mail.nedu.edu.cn; lesmatines@163.com

Abstract**—**The process of smart gird operation and management produces big volume of isomeric and polymorphic data, known as power big data. Since the data has a property of panoramic, the combination of data visualization and 3D scene is an efficient solution to comprehensible analysis and demonstration, while the highly detailed 3D models challenge rendering performance and the speed of human-machine communication. Therefore, this paper proposes a fast scene constructing method aiming at scenes that are composed of electrical equipment models. This method is designed based on a weighting function, which takes several factors that contribute to the complexity of a model into account. First, we select several specific factors of an edge. These factors are special because their values affect the surface of electrical equipment 3D models a lot, while affect common 3D models not obviously. A proper scenario is designed to quantize these factors and their corresponding weighting parameter. Second, in order to ensure the weighted contribution of each factor is balanced, we adjust the weighting parameter of each factor by restricting the range of the parameter. At last, we use two stacks for one model to record the edges that are to be optimized and have been optimized in sequence, sorted by the contribution value. Experiment shows that method in this paper meets the demand of power big data visualized analysis system by finely retaining figuration and high rendering performance.

Index Terms**—**Power big data, visualized analysis

I. Introduction

Research has shown that human eye has a far more sensitivity to graphic rather than data. Difficulties in comprehension increase as volume of data increases, meanwhile visualization is in need to be an assistant method. Visualization is proved to be an effective way in big data analysis, and is widely applied in practice. Nowadays the study of power big data mining prospers, while the subject of demonstrating the result of big data mining in a comprehensible form and of suggesting further strategies by the analysis result still remains a challenging issue. Power big data comes up with complex forms like space-time, dynamic and related data, and they are different from data of visualization in scientific computing like specific vector and tensor field. Besides, excessive overlap and density emerge when traditional visualization method is applied on power data, which is high dimensional and of massive amount. This situation leads to a less effective visualization result that lacks the visualized expression of relations between data. The essence of traditional 2-D visualization method is an abstract graphic symbol system, which condenses N dimensional data to 2-D. When N increases to a certain extent, the complexity of 2-D visualization like parallel coordinates graph actually weakens the convenience of human-computer interaction, and goes against with the original idea of enhancing comprehension of data.

Power big data is a kind of isomeric status data. It is categorized into operating data, business data and enterprise management data, all feature an attribute of 3D spatial topology, which is a suitable circumstance for 3D scene visualization. The Combination of power big data and 3D scene makes data no longer isolated and less abstract, therefore relations between data is revealed in spatial scale. It is clear that certain data is generated by its related entity, which could be an equipment, user or something else. This combination amends traditional visualization method by involving spatial dimension.

Traditional 3D virtual reality application has a slow performance in scene constructing because of its fidelity to real shape, lighting and texture. While in the situation of big data, 3D scene is required fast constructing and refreshing in order to meet the need of human-machine interaction, fast roaming and dynamic data visualization. The complexity of model is a main factor that contributes to the speed of scene construction. Complex models have more vertices, and it takes a large amount of calculation in vertex coordinate transformation. Power equipment scene consists of models of transformers, insulators, breakers and so on. These models have a lot of coil structures that make local surface curvature steep, and the curvature describes feature so it cannot be over simplified, while over retaining also makes it less efficient in scene optimization. This paper proposes an algorithm to simplify model efficiently. The algorithm quantizes factor that contributes to total complexity and weigh them in statistical method, then calculate the overall contribution of those factors in a way of balancing construction speed and fidelity. A method is involved in the implement of this algorithm, which enables lossless reconstruction.

Manuscript received May 25, 2015; revised September 23, 2015.
Corresponding author email: lesmatines@163.com.
doi:10.12720/jcm.10.10.773-777

II. MODEL OPTIMIZING ALGORITHM BASED ON WEIGHTING FUNCTION

Level of Details (LOD) is put forward by the phenomenon that surface precision could be reduced without effecting human eye vision when the model is far from viewpoint. LOD method gradually optimizes model by simplifying surface elements in levels, therefore reduces the total complexity of 3D scene. Fig. 1 shows different levels of a same model applying LOD method.

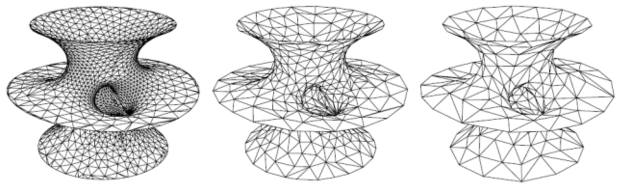

Fig. 1. model applying LOD method.

The mesh topology of a model has its mathematical representation as *M*=(*K*, *V*), where *K* is the connectivity between vertices, like edge or face; and *V* is vertex. The algorithm should process *K* in order, that is, less important ones first and important ones after. In order to quantize importance of *K*, many factors like edge length and triangle square should be included. This paper designs a function *f*(*K*) to quantize importance that each *K* contributes to model complexity.

After quantization, *f*(*K*) is organized into a queue in ascending order, where the related *K* is simplified when optimization takes place. Simplification occurs from the head of the queue, and a proportion parameter decides how many elements in the queue should be simplified. As gap could probably appear if *K* is immediately removed, we collapse K instead of remove it, avoiding extra calculation in filling gaps. We collapse *K* by a transformation $K=(V_1, V_2)\rightarrow K=(V_1, V_1)$, where V_1 and V_2 are the two endpoints of a *K*(an edge).

It is obvious that the design of function *f*(*K*) makes up the core algorithm. As is described above, we consider *K* as edge and select four factors of an edge: length *L*, flag *F*, sum square *S* of bilateral triangles and curvature *C*, which is defined as angle of the two normals of bilateral triangles. Flag *F* is a sign to identify if *K* is a border or not.

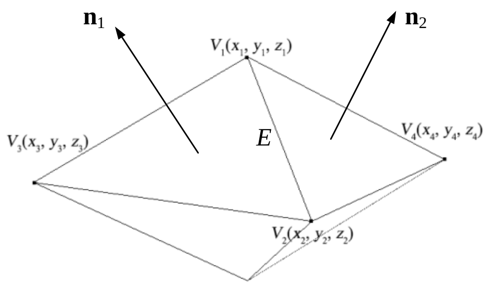

Fig. 2. Factors of an edge.

Fig. 2 shows edge $E=(V_1, V_2)$ has one triangle and its corresponding normal by each side. With coordinated of vertices, we quantize the four factors as follows:

$$L = \sqrt[2]{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}. \tag{1}$$

$$F = \begin{cases} 0, & V_4 \text{ does not exist} \\ 1, & V_4 \text{ exists} \end{cases}. \tag{2}$$

$$S = \frac{1}{2}\left\|\begin{vmatrix} i & j & k \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_1 & y_3-y_1 & z_3-z_1 \end{vmatrix}\right\| + \frac{1}{2}\left\|\begin{vmatrix} i & j & k \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_4-x_1 & y_4-y_1 & z_4-z_1 \end{vmatrix}\right\| \tag{3}$$

$$C = \cos^{-1}\mathbf{n}_1\cdot\mathbf{n}_2 = \cos^{-1}\left(\frac{\begin{vmatrix} i & j & k \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_1 & y_3-y_1 & z_3-z_1 \end{vmatrix}}{\left\|\begin{vmatrix} i & j & k \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_1 & y_3-y_1 & z_3-z_1 \end{vmatrix}\right\|} \times \frac{\begin{vmatrix} i & j & k \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_4-x_1 & y_4-y_1 & z_4-z_1 \end{vmatrix}}{\left\|\begin{vmatrix} i & j & k \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_4-x_1 & y_4-y_1 & z_4-z_1 \end{vmatrix}\right\|}\right). \tag{4}$$

Equation (5) defines basic weighting function W_0 for an edge. Weighting coefficients w_1, w_2, w_3 and w_4 are to be defined by subsequent equations.

$$W_0 = w_1\cdot L + w_2\cdot F + w_3\cdot S + w_4\cdot C. \tag{5}$$

Obviously, the range of *C* is restricted to $[0, \pi]$ and *F* is to $[0, 1]$, while range of *L* and *S* is not restricted, and their values vary according to the mesh structure. If we use the original value of *L* and *S* in calculation, there might emerge a huge error when it comes to adjusting the proportion of each factor contributes to the importance of *K*. Therefore we apply histogram equalization to four factors in order to linearly map all factor ranges to $[0, \pi]$. E_{L_i}, E_{F_j} and E_{S_k} are the mapped variables.

$$E_{L_i} = \frac{\pi L_i}{\max\ (L_1, L_2 \ldots L_n)} \tag{6}$$

$$E_{F_j} = \pi\cdot F_j. \tag{7}$$

$$E_{S_k} = \frac{\pi S_k}{\max\ (S_1, S_2 \ldots S_m)} \tag{8}$$

Now we have the new weighting function of factor ranges equalized. E_L, E_F and E_S are the mapped variables with restricted range $[0, \pi]$.

$$W = w_1\cdot E_L + w_2\cdot E_F + w_3\cdot E_S + w_4\cdot C. \tag{9}$$

It is apparently that short length makes it less important of an edge, therefore *L* with smaller value ought to be simplified first. With this consideration, we use standard deviation $\sigma_{E_L} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(E_{Li}-\overline{E_L}\right)^2}$ as weighting coefficient of *L*. Small σ_{E_L} represents that all edges have almost the same length, which means the length factor hardly causes difference in importance. Under this condition, the weight of *L* should be small, which is proportional to σ_{E_L}. We define $w_1 = \sigma_{E_L}$, and similarly w_3=σ_{E_S}, w_4=σ_C. Flag F tells if an edge K is a border(K has unilateral triangle) or not(K has bilateral triangles), of which former one affects topology a bit much comparing to the later one. Border edge makes the value of F to be 1, while non-border edge makes 0. As more border edges produce more distinct effect on model surface, we define coefficient $w_2 = \alpha$, which represents proportion of border edges among all edges.

In a conclusion, the importance of any edge could be quantized by a weighting function

$$W = \sigma_{E_L} \cdot E_L + \alpha \cdot E_F + \sigma_{E_S} \cdot E_S + \sigma_C \cdot C. \quad (10)$$

where every weighting coefficient has a range of $\left[0, \frac{\pi}{2}\right]$, and every factor has a range of $[0, \pi]$, and that assures every factor contributes a balanced value to the importance of *K*.

III. Constructing Data Visualization Scene

The optimizing process must be reversible to meet the purpose of lossless reconstruction in practice. After calculating of *W*, the scene is ready to be constructed. For each model's triangular mesh, we use a list L to store all pairs (E<V_i, V_j>, W) in ascending order sorted by *W*. Similarly, for each model's triangular mesh, we use a pair of stack S1, S2 to store edges to be collapsed and have been collapsed respectively. Initialize stack S1 by copying list L, while initialize stack S2 as empty. During scene construction procedure, whenever stack S1 pops an element, it is instantly pushed into stack S2, meanwhile the corresponding edge is collapsed. In reverse, stack S1 catches element that stack S2 pops during reconstruction procedure. Fig. 3 shows the procedure described above. An optimization proportion parameter controls how many edges should be collapsed at a certain circumstance, which is decided by simplifying level and the distance between the model and the viewpoint.

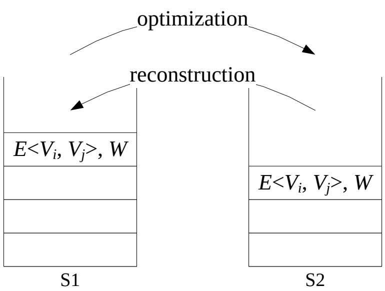

Fig. 3. Mesh optimization and reconstruction.

The scene construction procedure is executed as follows:

Step 1. Convert all model meshes into vertex arrays;

Step 2. For each vertex array, use a list L to store all pairs (E<V_i, V_j>, W) in ascending order sorted by *W*. Initialize stack S1 by copying list L, while initialize S2 empty.

Step 3. (Loop) Construct meshes. Collapse edges according to the optimization proportion parameter, in the meantime stack S1 and S2 transfer elements between each other.

Fig. 4 shows the whole procedure of scene initialization and construction, under the assumption that there is only one model in a scene.

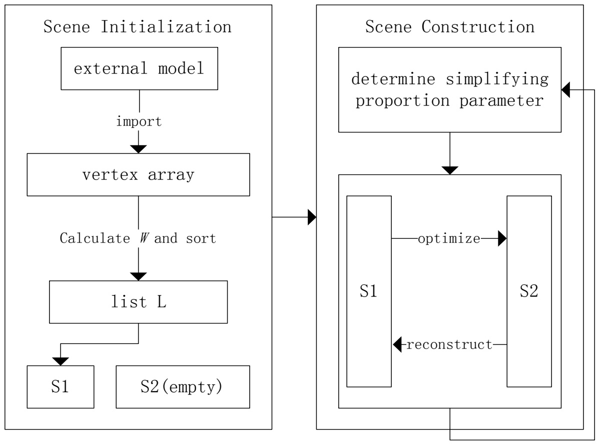

Fig. 4. Scene initialization and real-time construction procedure for one model.

The pseudo code of scene constructing process could be described as follows:

```
//scene initialization//
visualPowerData=visualize(powerData);
for(i=0;i<n;i++)
{
   //import all models then allocate s1, s2 for each one
   if(sceneObj.next!=NULL)
        mesh[i]=tri(sceneObj.next);
   Array w[]= weightFunc(mesh[i].e[]);
   List L=comb(w[],e[]);
   Stack S1[i]=sort(L);  Stack S2[i]=NULL;
}
//scene construction//
for(i=0;i<n;i++)
{
   prop[i]=calDist(mesh[i]);
   if(prop[i]>prop[i].pre) S1[i].pop(prop[i], S[2]);
   else S2[i].pop(1-prop[i], S[1]);
   render(S1[i]);
}
```

IV. Analysis of Instance

To measure efficiency and vision performance, we apply the method to a substation analyzing system. This system contains a scene of a 220kV substation, which in total has 536,141 vertices and 1,057,086 triangles. We use an optimization proportion parameter θ to control the number of vertices. θ means the proportion of vertices that is simplified in a model.

The algorithm is implemented by C++/OpenGL. Environment involves 8192MB memory, Intel i7-2760 QM CPU and NVidia Quadro 1000M graphic card.

Test is proceeded when θ equals to 0%, 70% and 85% respectively. We estimate result by considering average refreshing frame (measured with FPS, i.e. frame per second) and vision, with statistics and experimental data shown in Table I. The full detailed vision and optimized vision in different levels are shown in Fig. 5.

TABLE I: SCENE CONSTRUCTION PERFORMANCE

θ	Vertices	Triangles	FPS	Vision
0%	536141	1057086	3.9	full
70%	160841	316452	15.5	almost full
85%	80421	148352	18	details lost

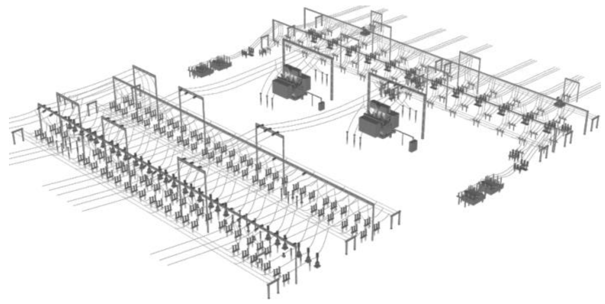

Fig. 5- Full detailed scene (θ=0%).

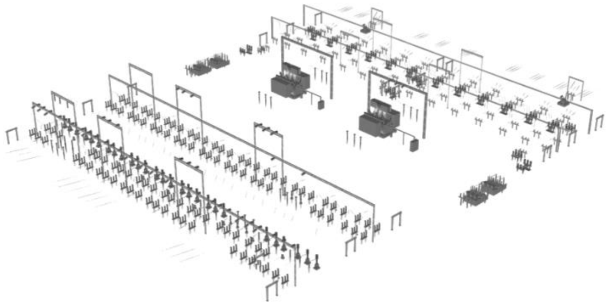

Fig. 5-b. Optimized scene (θ=70%).

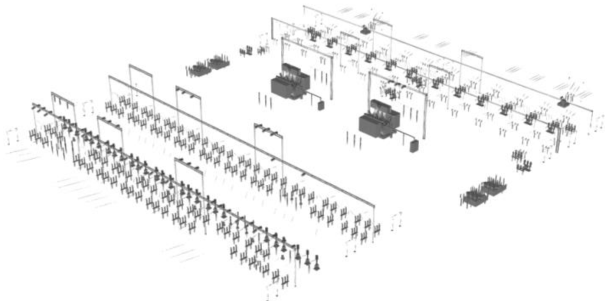

Fig. 5-c. Optimized scene (θ=85%).

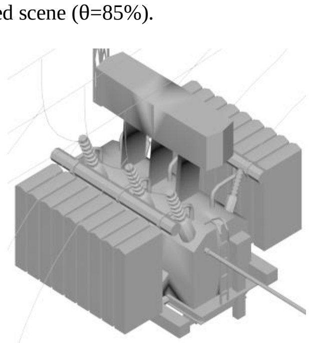

Fig. 6. Full detailed model (θ=0%).

A full detailed transformer model in orthogonal view is given in Fig. 6. We apply our algorithm and an algorithm with no boundary protecting mechanism on this model to compare figuration retaining performance.

When θ equals to 95%, the optimized result of our algorithm is shown in Fig. 7, algorithm with no boundary protecting mechanism in Fig. 8, both with wired model and rendered model provided. It is obvious that our algorithm retains more three-dimensional structures while loses insignificant details like wires, and algorithm with no boundary protection retains more insignificant details while collapses to two-dimensional structure in some part of the model because of lacking vertices. The result shows that algorithm presented in this paper is more efficient when dealing with electrical equipment which calls for local significant figuration retaining.

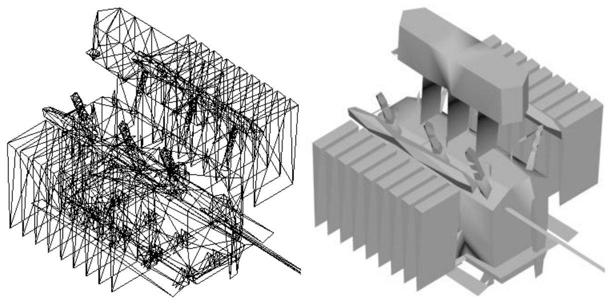

Fig. 7. Model optimized (θ=95%) by algorithm presented in this paper.

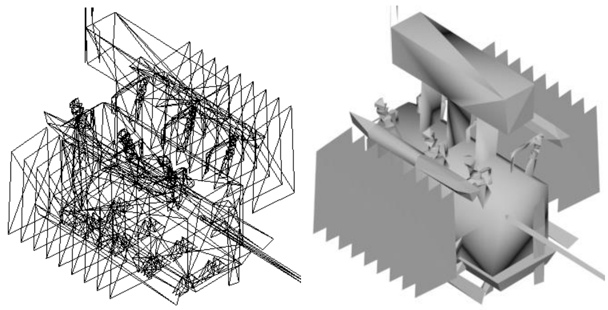

Fig. 8. Model optimized (θ=95%) by algorithm with no boundary protecting mechanism.

It is proved that the scene construction performance is considerably improved by applying the algorithm and method this paper illustrates, meanwhile vision retains finely with acceptable little changes occur. Constructed with original full detailed model (θ=0%), the scene could hardly run dynamically due to its low FPS (3.9); while a bit more details lost when θ=85%. Given the parameter θ =70%, system maintains an FPS of 15.5 with insignificant details lost, which meets the need of human-machine interaction, fast scene roaming and dynamic visualization of power big data visualized analysis system.

V. CONCLUSIONS

This paper proposes a fast model constructing algorithm based on weighting function aiming at power big data visualization, which accelerate scene construction by optimizing models in the scene. Optimization and reconstruction are opposite procedures that can be reverted to each other, thus enables lossless reconstruction. Experiment shows this algorithm is able to retain figuration finely and perform fast speed in refreshing frame in the application of power big data visualized analysis system, which requires capabilities of fast human-machine interaction, real-time scene roaming and dynamic visualization for data that changes with space and time.

We have already applied this algorithm and lossless reconstruction method to a visualized oil chromatography analysis system together with a 3D parallel coordinates graph, shown in Fig. 9. This visualized analysis system helps fast positioning and fault recognition by combining data and 3D scene instead of numerical tables and forms.

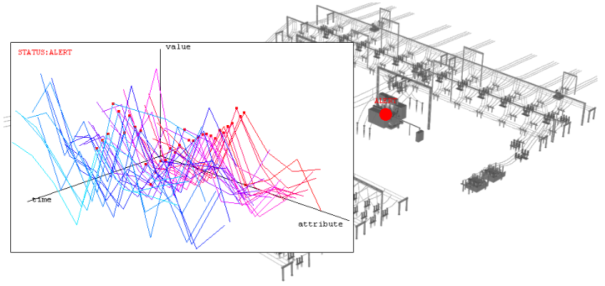

Fig. 9. A visualized oil chromatography analysis system

Since power big data has panoramic topology which can be properly integrated with 3D scene and data visualized analysis, such combination will be a future study area of power big data because of its advantage in enhancing human comprehension to the complex isometric data.

REFERENCES

[1] H. K. Chen, C. S. Fahn, J. J. P. Tsai, R. M. Chen, and M. B. Lin, "Generating high-quality discrete LOD meshes for 3D computer games in linear time," *Multimedia Systems*, vol. 11, pp. 80-94, May 2006.

[2] E. Feibush, N. Gagvani, and D. Williams, "Visualization for situational awareness," *IEEE Computer Graphics and Applications*, vol. 20, pp. 38-45, May 2000.

[3] P. C. Wong, H. W. Shen, C. R. Johnson, C. Chen, and B. R. Robert, "The top 10 challenges in extreme-scale visual analytics," *IEEE Computer Graphics and Applications*, vol. 32, pp. 63-67, Aug. 2012.

[4] D. Agrawal, P. B. E. Bertino, S. Davidson, U. Dayal, M. Franklin, *et al*. (Dec. 2012). Challenges and opportunities with big data - A community white paper developed by leading researchers across the united states. [Online]. Available: http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf

[5] A. Pregelj, M. Begovic, and A. Rohatgi, "Quantitative techniques for analysis of large data set in renewable distributed generation," *IEEE Trans. on Power Systems*, vol. 19, pp. 1277-1285, Mar. 2004.

[6] D. S. Wan and H. T. Ni, "Research on triangular mesh model dynamic simplification based on cost function," *Computer Engineering and Applications*, vol. 45, pp. 204-206, July 2009.

[7] T. Yan, "Triangular mesh simplification based on gauss curvature," *Computer Engineering & Science*, vol. 34, pp. 126-129, Dec. 2012.

[8] Y. Q. Song, G. L. Zhou, and Y. L. Zhu, "Present status and challenges of big data processing in smart grid," *Power System Technology*, vol. 37, pp. 927-935, Apr. 2013.

[9] J. Y. Liu, X. D. Shen, L. F. Tian, J. H. Chen, Y. Huang, and C. X. Li, "Prospects of visualization under smart grid," *Electric Power Automation Equipment*, vol. 30, pp. 7-13, Jan. 2010.

[10] J. M. Zhang, H. Chen, J. Chen, M. L. Zhou, X. D. Zhuang, and Y. J. Chen, "Smart grid situation awareness diagram modeling and conceptual design of situation awareness visualization," *Automation of Electric Power Systems*, vol. 38, pp. 168-176, May 2014.

[11] K. Yang, Q. Luo, and J. Y. Shi, "Parallel scatter plots: Visual analysis with GPU," *Journal of Computer-Aided Design & Computer Graphics*, vol. 20, no. 9, pp. 1219-1228, Sep. 2008.

Zhao-Yang Qu received his MSc and PhD degree in computer science from Dalian University of Technology, China in 1988 and North China Electric Power University, China in 2012, respectively. His research interests are in artificial intelligence, machine learning and data mining. He is now a professor of Northeast Dianli University.

Jing-Yuan Huang received her Bachelor degree in computer science from Nanjing University of Information Science and Technology, China in 2011. Her research interests are computer graphics, virtual reality and visualization in scientific computing. She is now a postgraduate of Northeast Dianli University.