GRAPHICS-IMAGE MIXED METHOD FOR LARGE-SCALE BUILDINGS RENDERING

Zhou Yang[1], Xu Qing[1]，Xing Shuai[1]，Hu XiaoFei[1]

[1.]Institude of Surveying and Mapping, Information Engineering University, ZhengZhou, China-zhouyang3d@163.com,13937169139@139.com,13838112822@139.com,huxiaofeicn@foxmail.com;

Commission II, WG II/4

KEY WORDS： City 3D， Graphic and Image Mixed， Large-scale Scene， Huge Data， Urban Building

ABSTRACT:

Urban 3D model data is huge and unstructured, LOD and Out-of-core algorithm are usually used to reduce the amount of data that drawn in each frame to improve the rendering efficiency. When the scene is large enough, even the complex optimization algorithm is difficult to achieve better results. Based on the traditional study, a novel idea was developed. We propose a graphics and image mixed method for large-scale buildings rendering. Firstly, the view field is divided into several regions, the graphics-image mixed method used to render the scene on both screen and FBO, then blending the FBO with scree. The algorithm is tested on the huge CityGML model data in the urban areas of New York which contained 188195 public building models, and compared with the Cesium platform. The experiment result shows the system was running smoothly. The experimental results confirm that the algorithm can achieve more massive building scene roaming under the same hardware conditions, and can rendering the scene without vision loss.

1 INTRODUCTION

Constructing a smart city is an effective way to achieve the sustainable development and enhance the comprehensive competitiveness of cities. As an important part of smart city, Virtual City has application value in many fields, such as military, urban planning, traffic navigation, social entertainment and so on. 3D City simulation technology application in the government departments of the municipal planning, can save design cost, improve working efficiency; In the military field, Training in the virtual city can improve the military combat effectiveness without the restriction of the political environment.

Urban model data is huge and complex structure, lack of an efficient visualization system is often one bottleneck of the digital cities application. With the progress of Technology, especially the rapid development of laser radar, unmanned aerial vehicle, oblique photograph, we can build a large scale textured 3D city models in acceptable time. The modeling efficiency and the quality of data are constantly improved, the technical difficulty of the visualization of large-scale urban 3D scene is also increasing. Although the capabilities of computer graphics hardware are greatly improved in recent decades, there are still many problems in the practical application of large-scale urban building model drawing technology. The 3D city simulation technology is one of the hot topics of common concern in GIS, 3D simulation fields.

2 RELATED WORKS

In the whole city scene, each visual scene contains thousands of buildings and millions of triangular patches, and because of the limited hardware performance of the computer, even with a complex algorithm, it's quite difficult to make full real-time rendering.

In the aspect of huge data storage service, the object oriented database which supporting spatial query is one of the mainstream schemes, the unstructured NoSQL database has become a hot spot of research. Shaojun L I provides technical details of a three-dimensional spatial database based on NoSQL(Shaojun L I,2014), Jung M G contrasts two common spatial databases of MongoDB and PostgreSQL(Jung M G et al., 2015). Spatial index is the key to efficient processing of massive data. The R tree is a widely used spatial index structure for massive urban 3D building models. LOD with Out-of-Core are the main methods of rendering the large scale 3D scene. Peng C implements the edge folding simplification algorithm, dynamic LOD technology by using GPU, and realizes the efficient rendering for complex models(Peng C, 2012). Jie H takes full advantage of GPU performance, implements R tree index by CUDA platform, and adopts a dynamic simplified LOD method in GPU to achieve efficient rendering for massive urban buildings(Jie H,2012). Zhou S propose a hybrid primitive LOD algorithm. The point and line primitive which is simpler used to build the coarse resolution model to improve the rendering efficiency(Zhou S, 2012). Compared with the traditional geometry LOD algorithm, the algorithm has multiple rendering acceleration. The R tree index is used and the adaptive LOD control method is adopted to realize the real-time loading for more than 20 thousand buildings(Gong J,2011). Based on the construction of network data service which accords with OGC standard, Prandi F implement it on the Nasa Java World Wind (NWW) and Cesium(Prandi F, 2015). Pispidikis I uses a service oriented visualization scheme for urban 3D model(Pispidikis I, 2016).

IBR(Image-based Rendering) is one of the mainstream technologies of large-scale urban 3D scene visualization. Compared with the geometric rendering technology, image rendering can drawing the large scale scene with unrelated to the complexity of the geometry. Street view map is a kind of widely used image rendering technology. However, some technical defects, like the lack of model’s geometric information and difficult to roaming continuous, restricting the IBR applications.

Most existing rendering techniques, especially level of detail (LOD) processing, devoted to reduce the amount of data drawn in each frame. The algorithm are very complex. We present a graphics-image mixed approach, firstly, split the view frustum into three regions: the interesting region, the less interesting region and the uninteresting region, then we adopt the different rending method in each region.

3 METHODOLOGY

In vision principles, the brain is just focusing on few objects in the scene, which is the region of interest. It is same in 3D city simulation, a object’s interesting level decrease as the distance becomes far away. So we can split the view frustum into three regions along the z axis: the interesting region, the less interesting region and the uninteresting region from near to far. Adopt different rending method in each region, make sure the real-time rendering and interactive in the interesting region. In the less interesting region and the uninteresting region, drawing the scene until the viewport is not moving. The IBR(image based rendering) and off-screen rendering technology is used, drawing the building models on a texture image which attaching the render buffer. At the end of a frame, blending the texture image with the interesting region, assemble the whole scene. All of the models only drawn once in the less interesting region and the uninteresting region, which can improvement in performance significantly. We also use the Frame-to-Frame coherence(Peng C, 2012). The main process of the method includes building data model, view split, data IO and scene drawing.

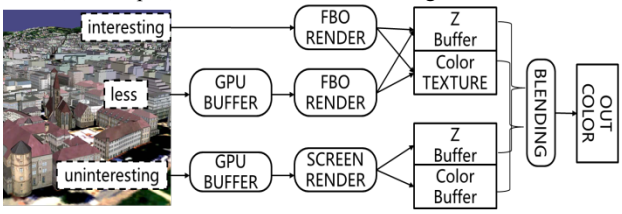

Figure 1: The overview of our approach

3.1 BUILDINGS MODEL

CityGML is an open data model and XML-based standard for the representation of 3D urban objects, it is therefore used at the data level to store the information on the Database, however, in spite of it is designed to represent 3D city models, it is not efficient to present or visualize 3D city models directly. For this reason, we need to reprocess the data. R-tree is one of the most widely used spatial index structures, it is unjustified that building R-Tree index for these models directly, the tree is too deep and it makes too many draw calls which will cost efficiency. Cesium is the 3D platform for visualizing globes, from every satellite in space to every building, their biggest initiative is 3D Tiles Streaming, an open specification for streaming massive heterogeneous 3D geospatial datasets, including terrain, imagery, point clouds, and vector data. We use the 3D Tiles method to build 3D Building Tiles, split the city space in a sparse grid layout, these buildings in a same grid make a tile, each tile is one OpenGL draw call. Also build 3D R-Tree index for these tiles, no more details for R-Tree index is presented here.

3.2 FRUSTUM SPLIT

The scene split method is a key in this paper, PSSM(Parallel-split shadow maps) is an effective method to generation large scale scene shadow. We apply it’s split method, split the view frustum into three regions: the interesting region, the less interesting region and the uninteresting region, and there will be two split points. It is very important to find the suitable points. The farther the distance from the split point to the near clip plane, the better the visual effect, but the performance is lower.

In order to balance the performance and visual effects, an adaptive method is adopted in this paper, setting appropriate split parameters based on the GPU’s rendering and storage capabilities. In the search stage, sort the nodes$[1, 2, \dots n]$ which need load data by the distance to eye point, the node k_1 and k_2 are the two split point, the interesting region is the nodes$[1, 2, \dots k_1]$, the less interesting region is the nodes$[k_1 + 1, k_1 + 2, \dots k_2]$ and the uninteresting region is nodes$[k_2 + 1, k_2 + 2, \dots n]$. Discuss the value of k_1 and k_2, the more the value of k_1, the better the rendering quality, while it will cost much time to draw the interesting region. Models in the less interesting region and the uninteresting region stored in GPU, the value of k_2 will refer to the GPU’s memory capacity. The value of k_1 and k_2 influenced by the model data and the hardware performance, assuming M_i is the maximum memory for the data, m_j is the memory size of node j, the split point k_i should be satisfied:

$$\sum_{j=k_{i-1}}^{k_i} m_j \le M_i \le \sum_{j=k_{i-1}}^{k_{i+1}} m_j \qquad (1)$$

At this point the value of the k_1 is just the initial value, during the drawing process, count the time used for real time rendering of interest region in each frame, optimize the value of k_1.

Assuming d_i is the distance from split point to the near clip plane, $d_i = dis(k_i) - near$, $dis(k_i)$ means the distance from node k_i to eye point.

3.3 GRAPHICS-IMAGE MIXED RENDERING

After split the frustum, adopt different rending method in each region. Rendering the models in interesting region on screen in each frame.

For the less interesting region and the uninteresting region, drawing the scene when user stop move mouse and the eye point does not move. IBR(image based rendering) technology is used, drawing the building models on a FBO texture image. We don’t clear the color buffer and depth buffer until it needs to redraw. After all draw calls are finished, blending the texture image with screen.

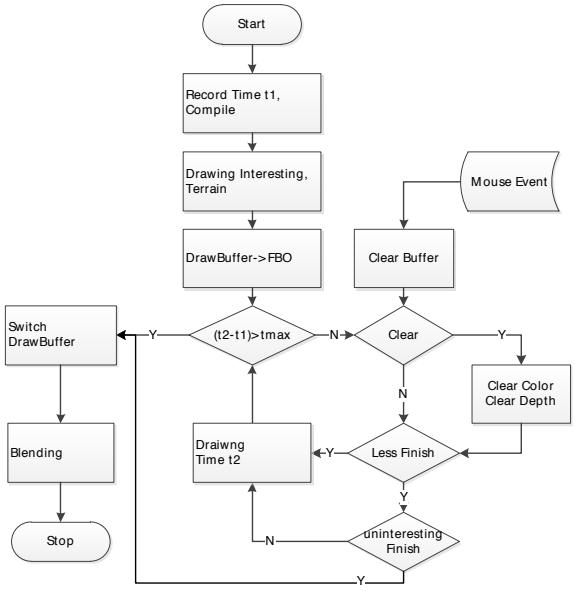

Figure 2: Graphics-image mixed rending pipeline

Update operation completed on the updated thread, and the drawing operation is only run on the drawing thread. The specific drawing process for each frame is shown in Figure 2: First, compile operation, compile the model data that has been loaded in memory to VBO objects. Then drawing the models in interesting region, record the costing time, the maximum drawing time minus the recording time is the time for rendering off-screen. In the off-screen stage, first draw the model data in less interesting region. Draw the uninteresting region with streaming method, create a cache buffer L_3 in memory, and update data in L_3 based on view location. Assuming M_3 is the max size of L_3, check the number of L_3 after update one model, the updating operation will suspend while the number of L_3 is bigger than M_3. Rendering thread read model data in L_3 in turn, and builds VAO objects for drawing. Move the data in another buffer, and release by GC.

3.4 BLENDING WITH DEPTH INFORMATION

In each frame, after all draw calls are finished, the texture image should blend with screen, the traditional RTT technology blending the FBO color texture as a texture unit for a geometry in the scene. It can’t deal with the z-fighting. For example, there is a mountain in the scene, and there are two build in the FBO, one is in front of the mountain, the other is behind the mountain. The RTT technology can’t deal with that directly. For this reason, a new algorithm that blending the texture image with screen in consideration of the depth information proposed in this paper. Before the off-screen rendering, enable the color and depth buffer. Setting the same size with screen buffer, also the model view matrix, projection matrix and lighting parameters, to get a same image. After the drawing, we get the color texture T_{color} and depth textureT_{depth}. Use the GPU programming, pass the T_{color} and T_{depth} in GLSL as *Uniform objects*. For each pixel, compare T_{depth} and the screen depth bufferC_{depth}, the smaller one is final color, the out color for pixel (s,t) is:

$$RGB_{\text{out}} = \begin{cases} T_{\text{color}}(s,t) & (T_{\text{depth}} < C_{\text{depth}}) \\ C_{\text{color}}(s,t) & (T_{\text{depth}} \geq C_{\text{depth}}) \end{cases} \quad (2)$$

3.5 BUILDING DATA STREAM

Based on the view frustum split, search and load data in each region. The retrieval operations for each region are executed on different threads, improve the retrieval efficiency. The retrieval process starts from the root node and traverses the scene tree with top-down, depth first traversal order. The specific retrieval steps are as follows:

1) Setting the root node as current node.
2) Calculate the spatial relationship between the current node and the view frustum, if current node is outside frustum, go to step 6); if current node is inside frustum, go to step 4); otherwise go to step 3).
3) Current node’s children nodes$[C_1, C_2 \dots C_n]$, choose the $C_i = min\,(d_1, d_2 \dots d_n)$ as current node, d_i is the distance from C_i to viewpoint. Go to step 2).
4) If the current node has finish load process, stop retrieving and return. Otherwise use the method in step3) to select child node, if the child node is leaf node, go to step 5); if not, handle children nodes in turn and repeat this step.
5) If the current leaf node needs to load model data, generate a request and place it in the update waiting queue. After the data has loaded in memory, place it in compile wait queue, and mark the node has updated.
6) Collect all nodes that have been loaded under this node, and put them in the remove waiting queue, the application will release them later.

The three regions are completely fit, that will repeat compute the nodes between two regions. For the leaf node, when calculate the spatial relationship with the far plane, only if the node is completely inside.

3.6 LOD

Multilevel detail is an effective method for large scale scene rendering. CityGML defines the detail hierarchy of the 3d city models, the building model is class into five different levels: LOD0 is an overview for the city; At LOD1, buildings are represented by block model with flat roofs; At LOD2, buildings have differentiated roof structures and thematically differentiated surfaces; LOD3 denotes architectural models with detailed wall and roof structures, balconies, bays and projections; LOD4 completes a LOD3 model by adding interior structures. LOD method applied in this paper, use the viewpoint distance as the LOD parameter, different level nodes correspond to different distance thresholds. Specific methods use the SSE(Screen Space Error):

$$d_i = \frac{h*k}{2r_i*tan\,(fov/2)} \quad (3)$$

In the formula, d_i is distance parameter of node in level i, fov is the field, h is the height of screen, r_i is node’s pixel parameter and k is the control coefficient.

4 IMPLEMENTATION

The algorithm is tested on a huge public CityGML model data in the urban areas of New York, which contains 188195 building models. The test system was deployed on a PC with Intel i5 processor, 8G memory, NVIDIA GT940 graphics card. The development environments were Windows 7 64 bit operating system, visual studio 12 compiler, Qt 5.4.2 and OpenSceneGraph-3.4.0.

The open source geo-database 3Dcitydb is used in the data processing work. 3Dcitydb is implement by PostgreSQL and PostGIS, which can extract every single building to a PostGIS database from the very huge CityGML file, and then export the buildings with KML format. The C++ programming language and OSG is used to process the KML file and build the R-Tree index.

According to the method of this paper, the frustum is split into the interesting region, the less interesting region and the uninteresting region. Different regions are distinguished with different colors, as the interesting region is brighter and the less interesting region and the uninteresting region is dark, as shown in Figure 3.

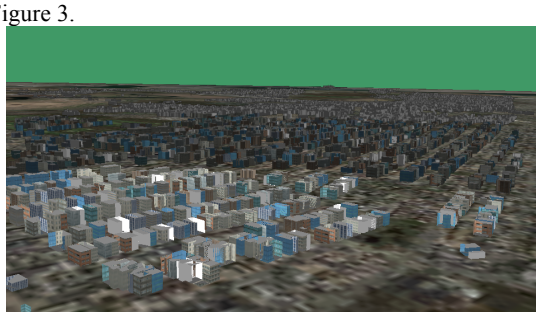

Figure 3: The scene split result

Integrated with the algorithm proposed in this paper, a large-scale building 3D scene roaming application with texture

information is development. Figure 4 are the rendering result for Manhattan.

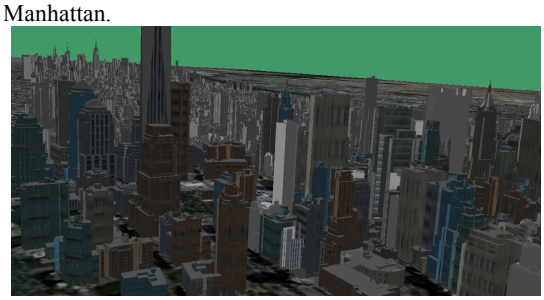

Figure 4: Building scene

Frame rate is one of the important parameters of the 3D simulation system, and the frame rate of this algorithm is tested. Several groups of experiments are conducted to count the scene drawing frame rate, and the amount of building model data is different in each experiment. The tests compare our algorithms with the traditional algorithms which used LOD and Out-of-core algorithms reduce linear.

buildings / method	10000	20000	40000	80000
traditional	24	13	6	——
ours	20	22	21	20

Table 1: FPS statistics

The result shows that the frame rate of this method is better than 20 frame/s in every scene drawing experiments, and frame rate is stable in each case, no matter how many building models in the test scene.

The Cesium 3D platform provides an open access service for the experimental data. We also compared the visualization result with Cesium platform, and the results shown in Figure 5 and Figure 6. Figure 5 is the result of our algorithm, as Figure 6 is the result of Cesium. The result shows that the LOD algorithm used in Cesium extract only some buildings in the distance to improve the rendering efficiency, which will lose some models in the scene. But our technology can drawing the whole scene without any loss.

Figure 5: The result of our algorithm

Figure 6: The result of Cesium

5 CONCLUSION

The algorithm proposed in this paper mixed the graphics rendering and the imaged rendering method. This algorithm maintained the continuity of scene roaming, and at the same time it has the advantage of image-based rendering technology that the rendering frame rate is independent of the scene scale. The experimental results demonstrate that, for the huge low resolution building scene, the algorithm can improve data bearing capacity of visualization system, especially the large-scale lower resolution building scene .It can roaming the huge building data smoothly under the conditions of that the hardware has lower performance relatively, and finally rendering the scene without vision loss. There are also some shortcomings in this method, the moving of viewpoint will make scene redraw. It will take some time to rending the off-screen image with a negative impact on user experience. We are going to do further research on the optimization: during the viewpoint moving, we can apply the affine transformation and resampling process on the off-screen image by the model matrix.

REFERENCES

Baig S U, Rahman A A. Generalization and Visualization of 3D Building Models in CityGML[J]. Lecture Notes in Geoinformation & Cartography, 2013:63-77.

Chan S C. Image-Based Rendering[J]. Computer Vision, 2014, 23(3):392-399.

Gong J, Zhu Q, Zhang H, et al. An adaptive control method of LODs for 3D scene based on R-tree index[J]. Acta Geodaetica Et Cartographica Sinica, 2011, 40(4):531-534.

He Z, Kraak M J, Huisman O, et al. Parallel indexing technique for spatio-temporal data[J]. Isprs Journal of Photogrammetry & Remote Sensing, 2013, 78(4):116-128.

Jie H, Tingyan X, Xiaoping R, et al. LOD methods of large-scale urban building models by GPU accelerating[C]//Proceedings of 2012 2nd International Conference on Computer Science and Network Technology. Changchun, China: IEEE, 2013:853-858.

Jung M G, Youn S A, Bae J, et al. A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment[C]//Proceedings of the 2015 8th International Conference on Database Theory and Application (DTA). Jeju Island, Korea: IEEE Computer Society, 2015: 14-17.

Kang S B, Li Y, Tong X, et al. Image-based rendering[J]. Foundations and Trends® in Computer Graphics and Vision, 2006, 2(3): 173-258.

Leite F, Akcamete A, Akinci B, et al. Analysis of modeling effort and impact of different levels of detail in building information models[J]. Automation in Construction, 2011, 20(5): 601-609.

Li L, Shen HW. Image-based streamline generation and rendering[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 13(3):630-640.

Li X, Wan W, Zhu M, et al. A fast real-time rendering method of 3D terrain using out-of-core visualization[C]//IET International Conference on Smart and Sustainable City(ICSSC 2011). Shanghai, China: IET, 2011: 1-5.

Patrick Cozzi. Introducing 3D Tiles[EB/OL]. 2015-08-10.https://cesium.com/blog/2015/08/10/introducing-3d-tiles/.

Peng C, Cao Y. A GPU-based Approach for Massive Model Rendering with Frame-to-Frame Coherence[J]. Computer Graphics Forum, 2012, 31:393-402.

Pispidikis I, Dimopoulou E. DEVELOPMENT OF A 3D

WEBGIS SYSTEM FOR RETRIEVING AND VISUALIZING CITYGML DATA BASED ON THEIR GEOMETRIC AND SEMANTIC CHARACTERISTICS BY USING FREE AND OPEN SOURCE TECHNOLOGY[J]. Isprs Annals of Photogrammetry Remote Sensing & Spatial Informa, 2016.

Prandi F, Devigili F, Soave M, et al. 3D web visualization of huge CityGML models[J]. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2015, 40(3): 601.

Schilling A, Bolling J, Nagel C. Using glTF for streaming CityGML 3D city models[C]//Proceedings of the 21st International Conference on Web3D Technology.Paris, France: ACM, 2016: 109-116.

Shaojun L I, Yang H, Huang Y, et al. Geo-spatial Big Data Storage Based on NoSQL Database[J]. Geomatics & Information Science of Wuhan University, 2017, 42(2):163-169.

Zhang F, Sun H, Xu L, et al. Parallel-split shadow maps for large-scale virtual environments[C]// Vrcia 2006 ACM International Conference on Virtual Reality Continuum and ITS Applications. Hong Kong, China: ACM, 2006: 311-318.

Zhou S, Yoo I, Benes B, et al. A hybrid level-of-detail representation for large-scale urban scenes rendering[J]. Computer Animation & Virtual Worlds, 2014, 25(3-4):245-255.