Research on the accelerated Ray Tracing for Volume-rendering based on GPU

Zhanfang Chen
Department of Computer
Changchun University of Science and Technology
Changchun, China
Email: chenzhanfang@cust.edu.cn

Yu Miao, Weili Shi,Tao Ren, Guoyu Zhang
Department of Computer
Changchun University of Science and Technology
Changchun, China
Email:jeffy2010@126.com

***Abstract*—Ray Tracing Algorithm is a main algorithm for volume-rendering in photorealistic rendering. This paper designs the algorithm based on stream for GPU,after analyzing the priciple of the Ray Tracing on GPU. According to the speed in modeling intersection being slow, realized the improvement of the algorithm on modeling intersection and scene ergodicity. With the result, It was found that the algorithm on GPU is better than on CPU.**

Keywords- GPU;Ray Tracing;accelerate; volume-rendering

I. INTRODUCTION

Ray Tracing Algorithm is a classic algorithm in volume-rendering technology, In traditional algorithms, all the calculations are operated on CPU with a slow reconstruction speed.So the real time effect is hard to achieve. In this paper, it adopts GPU programming technology. It puts the scenes intersection and ray ergodic and other steps into the GPU for processing instead of CPU. Through GPU's high speed of floating-point, it realizes 3-D the real-time volume-rending. Compared to ray tracing volume-rendering algorithm of CPU, the result in photorealistic rendering has been improved greatly.

II. GPU AND ANALYSIS OF RAY TRACING TECHNOLOGY

GPU is the abbreviation of Graphic Processing Unit. GPU is the core of the display card. It helps reduce display card's dependence on CPU and handle some of CPU's work with the floating-point calculation ability specially reflected when dealing with 3-dimensional images. The executive mode of GPU orders is different from that of CPU since the former can not realize recurrent directly, so the algorithm realized in CPU can't be executed directly. According to GPU's characteristics, this paper puts ray tracing in the form of CPU steam computing model, abstracts programmable GPU to stream processor and makes full use of the good function of GPU's parallel processing system structure to realize the algorithm.

The biggest problem to realize ray tracing based on stream is how to project ray tracing to the stream calculation mode. GPU has a strong ability to process stream, but has difficulty in the realization of recurrent. In this paper, it furthers divide the ray tracing into several kernels, which are connected through data stream. During the rendering of either static scene or active one, the first thing is to identify the type of the geometric body tablet element which the ray tracing can process and the accelerating structure type used in the system. Ray tracing can render those scenes which are made up of various geometric body tablet elements, and the triangle is the fittest. At the same time, image hardware can only support triangle rendering. Though other surface shape can be used, they are all transmitted into triangle before rendering. That's why it uses triangle to represent geometry in the scene. Then, the modes produced by modeling programm and scanning software are all made of triangle lattice. Ray tracing will be more simply and efficient when there is one kind of simple image unite in the scene. In terms of stream calculation, the same kernel set can process all the rendering in the scene, and this method simplifies the data stream of the system.

III. THE REALIZATION OF ALGORITHM

Ray Tracing Algorithm can be divided into two parts: one is achieved by CPU and responsible for the control and balance of the overall calculation, while the other part is working through GPU and this part which is in charge of the calculation of the ray tracing consists of 4 types of calculating kernel.

Plan from Jlin Provincial Education Department

A. CPU for Control

CPU is responsible for the over all control and calculating balance of the algorithm.

It will obtain a group of reflection light and a group of refracted ray when processing derivative technology in GPU. As the whole multipath calculation process is complex, the switching among those multipaths need the control of CPU. Since GPU can't calculate those rays simultaneously, it will need CPU's control over the distribution of the calculation resources, namely CPU will distribute GPU's calculation resources after the rays which are produced directly or by derivation are pressed into stacks.

Besides, ray ergodicity and intersection have become the toughest problem because of the limitation on visiting the storage space. Therefore, it is necessary to design a proper ergodic accelerating structure to organize the geometric primitive data in the scene. There are two organizing methods in classic accelerating structure. One is centered on the primitive of the scene while the other is centered on the space of the sences. This paper employs the latter method to construct algorithm similar to BSP. Meanwhile, in order to improve efficiency, It adopts the way that a group of rays paralleled into stacks.

B. Algorithm's Realization On GPU

A ray tracing algorithm based on stream is used in GPU ray calculation. The whole frame is made up of 4 functional kernel: ray generation, scene ergodicity, modeling intersection, rendering and light derivation. Each relates to the calculation of a path in CPU.

Ray generation: generate the start point and the direction of the original ray.

Scene ergodicity: use binary tree like structure to organize the scene and seek surface primitive of modes which may need the intersection of rays.

Modeling intersection: mainly deal with the finding of intersection and the calculation of the surface primitive intersection.

Rendering and ray derivation: process light computing through phone lighting mode. If there is an intersection, produce related reflection ray or refraction ray with the intersection information. When it is necessary, shadow will be produced to test the ray.

As is shown in Figure 1, the input of each functional kernel can be displayed on the left of the box, and the type of

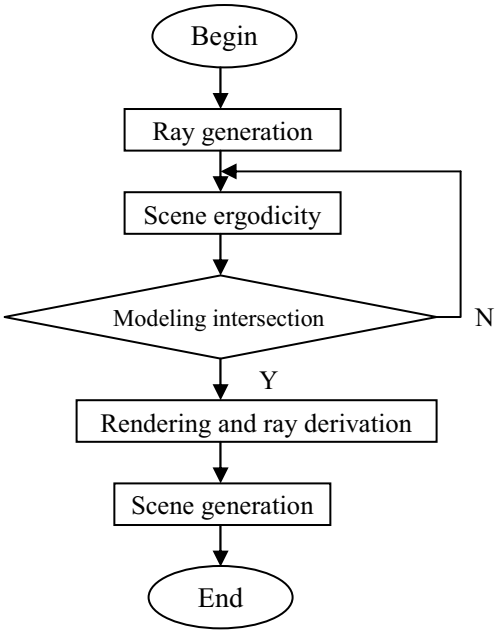

Fig.1 the ray tracing based on flow

the transmitted data stream between kernels is represented by the contend pointed by the dotted line. This dividing method is not compulsory in stream programming mode.

The kernel of the ray producer can generate a beam of ray stream, and each ray is related to a certain pixel in the image, so that the ray vector set is formed. The kernel of lattice ergodicity can read the ray stream produced by ray producer, then functions the ray to make it traverse the lattice step by step until finding a voxel which contains triangular faces. Thus, ray and voxel are output and transmitted into the kernel where they are intersected.

In scene ergodicity, the improved algorithm can be described as:

Firstly, establish 2-dimensional KD-Tree structure to store scene information. Every node in the tree stands for an axis-aligned bounding box and every inner node represents separation plane which divides the scene into two sub-regions. The way of carving out bounding boxes is cutting circularly along axis: first, divide the box from root node along x axis, then divide the second-level box along y axis, finally divide the third-level box along z axis, and it cycles by this means.

Secondly, change the order clue into threaded binary-tree before calculating intersection and ergodicity.

Thirdly, if the ray's span crosses two sides of the separation axles, that ray goes through two second-level boxes at the same time. In this case, we will first calculate its intersection with the first child node. Once the first node is traversed, the second node will not be pressed into stacks. Otherwise,, the intersection between the child node and the ray has to be figured out.

Fourthly, if that node is a leaf node, and the reflection has intersection with that node, the result will be a success. Otherwise, successor nodes of that node will be searched directly.The algorithm can avoid the expenses which are brought in by the stack operation. Though there are still expenses in establishing threaded binary-tree, it was all finished during the initialization and later operations will accelerate the speed of the ergodicity. In practical application, we limit the depth of the mediate node and ensure the balance of KD-Tree. The time complexity of the average algorithm is O(log2n).

Modeling intersection kernel is mainly in charge of testing whether the ray intersects with triangular faces which was contained in voxels. Ray tracing is actually the process of intersection between complex scene and the ergodicity of a ray tree. The new ray produced every time is the sub-ray of the original one, so that a ray tree is generated dynamically. The final color of every pixel point is obtained through the iterative intersection down the ray tree. Due to the limitation of GPU system structure, the ray tree can't be generated dynamically, but it can use ray stacks to control the sub-ray newly-produced by every intersection. In terms of reflection ray, it can use circular iteration to replace the original recursive transfer . In order to improve efficiency. The formula of N times iteration can be in the form as follows:

$$C(n) = \sum_{j=1}^{n} M_j \left(\prod_{i=1}^{j=1} r_i\right)(1 - r_i) \tag{1}$$

In this formula, Mj represents model j, ri stands for the reflection coefficient of object i, C(n) is the final color of pixel and N is the reflection depth.

The function of rendering and ray derivation is to figure out the color value. If a ray ends at the intersection point, the color value of the point will be written into the accumulated image. Besides, rendering kernel may generate shadow or sub-ray, when the newly-produced ray will be returned to the ergodic stage for new tracing.

IV. RESULTS AND ANALYSIS

Since the purpose of the experiment is to test what effect GPU acceleration will have on the speed of rendering, the test of the accelerated speed of ray tracing based on GPU doesn't contain the time cost in accelerating the construction of the structure. For the rendering of the static scene, speeding up structure construction can be accomplished in the pretreatment stage. It will analyse the accelerated ray tracing rendering based on GPU. In that algorithm, GPU achieves the application of CUDP(Compute Unified Device Architecture) proposed by NVIDIA in part of the process. as show in Fig.2.

According to the number of independent ALU in the GPU, we can separate ray projection operation. All the ALU will be computed in parallel, so that the processing speed of ray projection can be improved rapidly.

Fig.2 abdominal skeletal

The paper discusses the improved ray tracing algorithm, which is based on the GPU accelerated ray tracing rendering algorithm. According to the speed between the light and volume in modeling intersection being slow, realized the improvement of the algorithm on modeling intersection and scene ergodicity. With the result, It was found that the algorithm on GPU is better than on CPU.

[1] Yang Junhua,Fu Hongguang,Guo Hui.Design and Reality on Fast Ray Tracing based on GPU. COMPUTER APPLICATION[J]. 2007,27 (8):2033-2035.

[2] Chu Jingjun.Direct volume-rendering technology based on GPU. Master Degree thesis from Shanghai Jiao Tong University.2007.

[3] Zhang Shengjie,Wang Guorong,Wang Xiaoping,Research on accelerate Ray Tracing technology based on GPU. COMPUTER ERA.2007,6:36-38.

[4] Wang Biwei. Analysis of Ray Tracing algorithm based on GPU. Science & Technology Information.2007,23:6.