

Application of Automatic Analysis of Image Data Based on KD-Tree in Ray Tracing Technology

Weiming Zhao

Private Hualian College, Guangdong, Guangzhou, 510663, China

Abstract. In the field of computer graphics, ray tracing technology is widely used to generate high-quality "photo-level" images. The automatic analysis of KD-Tree image data is the data structure of dividing K-dimensional database space, and it has important application in the fast search of multi-dimensional space feature vector. Based on the automatic analysis of KD-Tree image data, the reasonable and effective application in ray tracing technology can make the verisimilitude of graphics reach the leap-type change, and realize the ideal application effect.

Keywords: KD-Tree, Image data, Automatic analysis, Ray tracing technology, Application strategy

Ray Tracing Technology is known as the Holy Grail of computer graphics, and is no stranger to engineers in the industry who design and simulate related products. In the real world, when light strikes the surface of an object, it bounces, scatters and bounces in a variety of ways until it finally reaches the eye. It is this complex interaction that creates the "horizon" of reality.

1 AUTOMATIC ANALYSIS OF 1KD-TREE IMAGE DATA

When calculating lighting for a scene, the Phong lighting model can be calculated using a fixed graphics rendering pipeline, which makes the rendered object look plastic due to the characteristics of the model. If you want to render an object that has a metallic texture and reflects its surroundings, the Phong model is useless. The programmable graphics rendering pipeline is much more powerful than the fixed rendering pipeline, although many realistic lighting effects can be achieved, such as the use of environmental maps to reflect objects to the environment. But this environmental reflection reflects only the images that are already saved in the Cube Map. In the real world, if an object that reflects its environment is surrounded by a lot of other objects, they will reflect each other. General environment mapping technology can not achieve this effect, so in rendering photo-level screen, we need to use ray-tracing technology.

As GPU performance is doubling every year, dynamic ray-tracing animation is becoming an industry trend. KD-tree and BVH-tree are the most commonly used methods for spatial partitioning of scenes. Both of these approaches can do ray tracing, but in dynamic scenarios, BVH-tree performs better than KD-tree. Meanwhile, experimental studies have shown that surface area heuristic SAH method can generate high-quality spanning trees, but it is time-consuming, which hinders the rapid construction of BVH-tree, and even though the SAH mechanism is optimized, it still can not achieve the effect of dynamic ray tracing. But if you build a BVH-tree on your GPU and calculate the SAH FUNCTION ON YOUR GPU, you can greatly speed up your ray-tracing preparation, moreover, using GPU parallel processing can make up for the low efficiency of CPU serial computing.

Kd-tree, K-dimensional tree, is a binary tree that stores some K-dimensional data. Constructing a Kd-Tree on a K-dimensional data set represents a division of the k-dimensional space of the K-dimensional data set, that is, each node in the tree corresponds to a K-dimensional Hyperrectangle. The Kd-Tree construction algorithm is mainly embodied in the selection of the most generous k dimension in the K-dimensional data set, and then the selection of the median m for pivot on that dimension to divide the data set. Get two subsets; create a tree node node at the same time for storage;

iterate over the two subsets until all the subsets can no longer be divided. When a subset can no longer be partitioned, the data in that subset is saved to the leaf node.

In general, KD-Tree creation algorithm is designed based on serial of CPU, and adopts data structure-preorder traversal method of stack. In single-core CPU, although the theoretical time complexity of KD-Tree serial creation algorithm is $O(N \log(N))$, the creation of KD-Tree still can not meet the requirements of many applications for complex set of elements. In this paper, the parallel performance of KD-Tree creation algorithm is optimized by fully improving the original algorithm and discovering the part of parallel computation. Compared with Quadtree/Octree, it is more suitable for models with different densities. Quadtree/Octree uses fixed-size grids to split the space, which increases the depth of the tree. For some of the space without the model also occupied the node, resulting in waste.

2 THE DEVELOPMENT AND APPLICATION OF RAY TRACING TECHNOLOGY

In traditional graphics rendering (rasterization), the lighting map and shadow map are pre-calculated and then applied to the scene to simulate the appearance of the scene. Ray tracing is a rendering technique that can accurately track the path of light in a scene, compared to rasterization, it can render reflections, Refractions, Shadows and lighting effects that are believable and have a higher degree of realism and fidelity. Ray-tracing simulates how light behaves in the real world, allowing for a leap in graphic realism to create 3-d scenes that are almost indistinguishable from real life.

On the mobile side, the cost of "spoofing" shadows and reflections in a smartphone game, such as a common game engine, has become prohibitive because of the traditional rasterization method, the reflection is generated using cascading shadow maps, which require rendering the geometry on the screen multiple times and writing the shadow map lookup table to memory, all of which consume cycles and bandwidth, and generate a lot of GPU and system power consumption. By bringing advanced 3D graphics processing capabilities to mobile devices, it creates a whole new capability for the mobile graphics processing industry to realize the vision of bringing lifelike graphics content to mobile devices, and drive a new revolution in smartphones. Ray Tracing Technology is a simulation of the human eye to receive light information, in the retina formation of the image process, according to statistics, the human eye received light information is limited, normally the human eye receives about 500 million pixels, meaning that the retina's image information can be broken down into 500 million very tiny lines of light and then traced back in the opposite direction, and detect the light corresponding to the information of the scene object.

Ray-tracing technology is based on this physical principle, in the process of generating computer images, the eye simulation abstract into the scene of the camera, the retina simulation abstract into the display screen or film, the 500 million rays are reduced to film pixels, and each pixel on the film is connected to a ray from the camera position to track the light at the point where the ray intersects the object in the scene. First of all, the camera film is subdivided into a grid, like the screen pixel point, the goal is to determine each pixel point lighting information. For each pixel, the light is emitted from the camera position and tracked to the pixel point. The beam of light is then tracked to see if it intersects with the object in the scene. If intersecting, the color value of the intersection point on the surface of the object is calculated. The color value of the point is recorded as the color value of the pixel. If not, the background color of the scene is simply filled into the current pixel, and then the next pixel is processed, looping until all the pixels have a value.

Compared with the traditional raster rendering technology, the ray tracing algorithm is more clear. Taking the viewpoint as the starting point, n rays are sent to the scene, then BXDF and BRDF are calculated according to the material of the collision point, and then diffuse reflection, specular reflection or refraction are carried out, this recursion cycle until the light escape or reach the maximum number of reflection, the final N light of the Monte Carlo integration can be obtained. The `clightsource` class has four member variables representing the position of the light source, the

ambient light component of the light source, the diffuse component, and the specular component. Similarly, all set and get methods provide the same functionality for subclasses of the class. Finally, there are also three virtual member functions, EvalAmbient, diffuse, and evaluular, whose names indicate their function, and all return values of type GVector3 -- color. Since the calculations may vary for different types of light sources, they are set up as virtual functions in preparation for future expansion. I put the lighting calculation in the light source type here, of course, you can also put it in the object class CGObject, or you can write a separate method that takes the light source and the object as parameters, the color is calculated and returned as the final return value.

3 THE APPLICATION OF AUTOMATIC ANALYSIS OF IMAGE DATA BASED ON KD-TREE IN RAY TRACING TECHNOLOGY

Compared with BSP, the difference is only in the way of segmentation and end conditions, the use after segmentation and BSP are exactly the same. Since the condition of the end of segmentation does not require the formation of convex polygons, the number of leaf polygons for rendering can be freely determined. To be able to generate a new generation of graphics hardware scene organization structure. In BSP, the non-foliated node preserves the segmentation surface, which is used for rendering. So you can't use BSP for dynamic scenes. The surface in the model is used as the segmentation surface, which will bring some uncertainty to the algorithm.

3.1 CPU and GPU

The computing power of heterogeneous parallel devices is mainly embodied in three aspects: thread concurrency, thread storage resources and thread computing resources. The on-site concurrency of CPU is very low, but each thread has a lot of storage resources and rich computing resources. This means that its algorithm is better suited to allocating data, and then each thread processes the data relatively independently, known as the MIMD algorithm pattern. So the CPU thread is usually said to be a relatively heavy field. But for GPU, because of its huge number of stream processors, it needs high thread burst to make full use of these stream processors. If the burst model can not reach the high burst, it is difficult to make full use of GPU's performance. But each thread has less storage resources and less computing resources. This means that the GPU thread is a relatively lightweight thread because it can not allocate data to threads and must be executed on a module by module basis, the SIMD algorithm pattern.

3.2 In multicore cpus -- multithreading

Through the way of thread, can be in dual-core, four-core CPU for a certain degree of parallelism. However, due to the limitation of the number of CPU cores, if we parallelize the fourth layer of KD-Tree in the same way, we will not get the ideal optimization effect. Because there are eight nodes in the fourth-level KDTree and eight threads need to be created (including the main thread) , the number of threads is greater than the number of physical compute cores of the CPU. In each cycle, only half of the threads are actually working, and the CPU threads need to switch frequently. The constant switching between threads greatly reduces the creation speed of KD-Tree.

3.3 In CPU and GPU -- multithreading and GPU features

The CPU is used to build the first few layers of KD-Tree. When the number of layers of KD-Tree can be parallel, the GPU is used to create the remaining layers of KD-Tree. Compared with the traditional algorithm, the parallelism of this algorithm is greatly improved, but its running speed is similar. There are two main reasons for this: in the process of switching from CPU to GPU creation, a large amount of memory data is transferred from main memory to video

memory. Because of PCIE's limited bandwidth, transmission will take a lot of time. The termination condition of the CPU creation algorithm is that the number of all nodes is less than a certain threshold. If the threshold setting is not reasonable, the CPU may stop running and individual threads of the GPU will traverse a large amount of data. Because the CPU contains a multi level Cache system, it can be used to hide the memory access delay, but the GPU is hidden by thread switching, and is idle, which wastes a lot of hardware resources.

4 CONCLUSION

Ray tracing algorithm is an important algorithm in the field of realistic graphics. It combines the color of light with the material properties of objects by simulating the reflection and refraction of light in the process of propagation, it can simulate complex 3D objects well and is suitable for rendering scenes with high Surface finish or transparent surfaces and mirrors. In the whole algorithm, the most computationally intensive is the intersection of light and scene graphic elements. If the dynamic ray tracing is considered, the whole scene space structure should be established.

REFERENCES

- [1] Shaodan Wang, Ni Li, Guanghong Gong. Research on real-time simulation method of infrared sensor frequency domain physical effects based on GPU. *China Stereology and Image Analysis*, 2015, 20(3): 252-259.
- [2] Jingsheng Wu, Xudong Bao. Image volume rendering based on GPU general computing. *Biomedical Engineering Research*, 2008, 27(3): 175-178, 192.
- [3] Jiayan Tan, Guofeng Liu, Jingyu Gao. Research on high-order finite difference forward modeling of seismic wave field based on GPU computing. *CT Theory and Application Research*, 2016, 25(1): 1-12.
- [4] Sen Si, Xinquan Wu, Shanglian Bao, et al. Dose simulation of Monte Carlo radiotherapy based on GPU. *Chinese Journal of Medical Physics*, 2015, 32(1): 1-3.
- [5] Heqi Lu, Peng Bao, Jieqing Feng. Real-time KD-Tree and dynamic scene ray tracing based on OpenCL [J]. 2013, (7).