2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

Global Illumination of Dynamic 3D Scene Based on Light Transport Path Reusing

Yan Ding[1] , Chao Xu[1], Hua Li [1]* , Xiaohan Sun[1]
1 School of Computer Science and Technology
Changchun University of Science and Technology
Changchun, China
lihua@cust.edu.cn

Yunbo Yang[2]
2 College of Engineering
Yanbian University
Yanji, China

***Abstract*—Interactive global illumination (GI) plays an important role in movie production and many applications generated by computer graphics. In this paper, we focus on a fast method to produce interactive GI by reusing the light transport path of static geometry in the virtual scene. Our method divides the 3D scene into two categories: static and dynamic geometric objects. The scenes are managed by two structures called SSG and DSG respectively. For the static objects, the transport path is reused between adjacent frames by a bidirectional light tracing strategy, which reduce the computing cost and the noise due to the randomness of sampled light sources. The experimental results show that the proposed method can reduce the computation time up to 30% compared with traditional path tracing.**

Keywords-global illumination; path tracing; ray-tracing; rendering

I. INTRODUCTION

Reproduction of the real world is an ultimate goal of computer graphics ever since it was introduced. Global illumination (GI) computes the light transport in virtual scenes with the reflectance of surfaces or that scattered or absorbed by the surfaces, and computes the visibility between a point and a surface, or two surfaces. Most of the interactive GI is obtained by triangle rasterization on extremely fast and increasingly sophisticated commodity graphics chips with an approximations simulation of real world. On the other hand, offline rendering is more wildly used in movie industries, animation for its physically-correct and high-quality rendering based on ray tracing and path tracing.

Path tracing is a Monte Carlo rendering algorithm that integrates the shooting and gather power to create photorealistic images. The idea is to sample the flux through the pixels, and gathering light from all the followed light paths back to the light source. In the dynamic scene, the movement of geometric object makes the light path changed between frames and flicker raised in terms of randomness sampling of Monte Carlo algorithm. Increasing the number of sampled light will produce more calculation that makes it impossible in interactive applications. Reusing part of the light transport paths will reduce the differences of adjacent frame and reduce the computational cost. In this paper, we propose a new method and structure that reuse the light transport path, which can solve the mentioned issues. Firstly, the method subdivide the geometry of the dynamic 3D scene into static parts and dynamic parts. All the static and dynamic geometric objects are organized and managed by two structures called SSG and DSG, which store the relative information of intersection of both sampled light source and view point sampled lights as well as the transport path. For the static sub-scene, the unchanged light transport path is reused by checking out the SSG information. This method is implemented by the combination of ray tracing from light source and path tracing from view point. The experiment results show that this approach can reduce computational cost and reduce the flicker and noise with respect to the same sampling resolution of dynamic 3D scene significantly.

The rest of this paper is organized as follows. In section II, We firstly introduced the principle of Monte Carlo Path Tracing and some improved algorithms, and then proposed a light transport path reuse algorithm with virtual object organization and management strategy of path tracing. We presented experimental results and analysis in section III, and in section IV, our method was compared with ray tracing algorithm.

II. CLASSIFICATIONS OF GEOMETRIC OBJECTS

According to the idea of Wald [1], the motion transformation of the dynamic 3D virtual scene is divided into three categories: 1) completely static state; 2) structured movement, including rigid body motion and affine transformation; 3) motionless structure. Our algorithm utilizes a combination strategy of ray tracing and path tracing to preserve the light from the unchanged light transport path of the static geometry in the scene.

A. Monte Carlo Path Tracing

Monte Carlo Path Tracing (MCPT) was introduced to computer graphics by Koivisto [2] in 1986 as shown in figure 1, which generate the solution of global illumination. The principle of the algorithm is as follows:

Firstly, it expresses the rendering equation as hemispherical area integral form and integrates radiance for each pixel by sampling paths randomly as equation (1).

$$L_0(x,\vec{w}) = L_e(x,\vec{w}) + \int_\Omega f_r(x,\vec{w}',\vec{w}) L_i(x,\vec{w}')(\vec{w}'\cdot\vec{n})d\vec{w} \tag{1}$$

Where $L_e(x,\vec{w})$ is emitted at a point x in direction $\vec{w}$, $f_r(x,\vec{w}',\vec{w})$ is BRDF of the surface, which reflects the

978-1-5386-3066-2/17 $31.00 © 2017 IEEE
DOI 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.44

character of surface material. $L_i(x,\vec{w}')$ is radiance received from direction $\vec{w}'$ and $\vec{w}' \cdot \vec{n}$ is the dot product of normal vector and incident light vector.

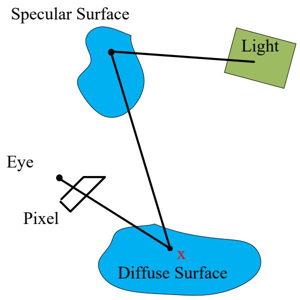

Figure 1. Monte Carlo Path Tracing.

And then from the starting point, we send a beam of light through the surface of each object and calculate the internal multiple interaction (reflection, refraction, scattering) to the end. The contribution value (the irradiance) is converted into an integral of the paths formed by these surface interaction points. Therefore, the problem of solving the rendering equation is transformed into the Monte Carlo integral calculation for the path sampling of the scene and the contribution value of each path is averaged. According to the difference between the starting point and the ending point, the path is divided into forward ray tracing and backward ray tracing. The forward path traces the light from eye point and propagates toward the light source, called eye tracking. In contrast, in the backward ray tracing, the light starts from the light source and is traced to the eye.

However, in order to be more accurate, the algorithm faces two theoretical and practical challenges: sampling of difference paths and low variance estimates. Each sample of the Monte Carlo methods is independent and it is difficult to sample the same path between different frames which prone to flicker aliasing. At the same time some sampled path requires a large number of samples in order to achieve plausible results, which resulting in the algorithm has a very slow convergence rate. Traditional variance reduction techniques such as importance sampling, hierarchical sampling, quasi-Monte Carlo methods still can’t change this convergence rate. Many efficient and classic algorithms are presented so far to accelerate the convergence. Vertex Connection and Merging and Unified Path Sampling (VCM/UPS) are effectively identical techniques independently developed by Georgiev et al. [3] and Hachisuka et al. [4] in 2012, respectively. VCM/UPS combines bidirectional path tracing and progressive photon mapping, which is particularly advantageous for specular-diffuse (SD) paths and specular-diffuse-specular (SDS) paths. Hachisuka et al. [5] fuse the underlying key ideas behind VCM/UPS and Markov chain Monte Carlo (MCMC) into a single, efficient light transport solution, especially in scenes with complex glossy or specular transport and complex visibility. Bidirectional Path Tracing (BDPT) [6] treats light sources and the viewing point on an equal basis. BDPT method builds eye sub-path and light sub-path and connects the interact vertices to estimate the contribution of a sample point [7] using the multiple importance sampling strategy [8]. Connecting each pair of eye and light vertices is an important part in BDPT. It has been attempted in combinatorial BDPT , in which the connections were off-loaded to the GPU. Walter et al. [9] proposed a Bidirectional Light Cuts (BDLC) algorithm which reduced the number of computed connections using a spatial data structure with respect to the last vertices of the sub-paths, and computed the contribution for the tree structure. Vorba *et al*. [10] proposed to guide the choice of the direction of the next ray at a path vertex by using parametric mixture to represent the distributions with online learning approach. Inspired by the BDPT, we combine the basic ray tracing and the path tracing together to reduce the test rays and reuse part of the light transport path by analyzing the character of geometries and the symmetry of Bidirectional Reflection Distribution Function.

B. *Light Transport Path Reuse Algorithm*

When the virtual scene is modeled, the geometric objects in the 3D scene are divided into static geometric objects and dynamic geometric objects. Figure 2 shows a dynamic scene generated by ray-tracing. The test model is Sponza and the number of polygon is 238395.

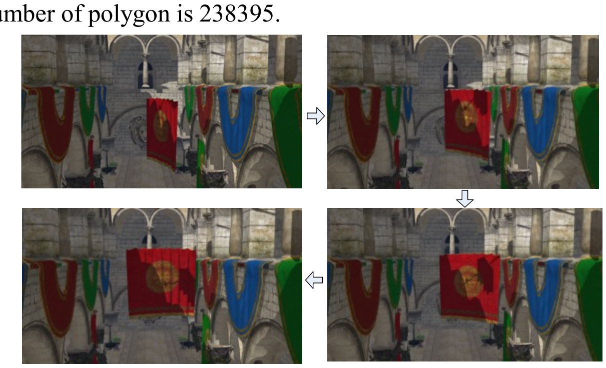

Figure 2. Dynamic scatter of different frames.

As this paper focuses on the hierarchical structure management and the reuse of inter-frame transport path to reduce the inter-frame flicker, so the deformation of the dynamic geometry is not taken into account. BVH accelerate structure is used to accelerate the interaction in dynamic scene.

C. *Virtual Scene Organization and Management*

The main steps of reusing the inter-frame light transport path method is shown as follows:

1) Generates and tracks a series of light rays emitted by the light source. In the SSG structure, the intersection of light rays and the surface of static geometry are stored as well as the points along with the transport path based on reversed ray tracing. The BRDF and the material properties determine the recursion depth of the ray tracing.

2) Generates the sampled sight rays from the viewpoint to the screen pixel, and calculates the first intersection of the

sampling ray and the surface of the scene. After determining the first visible point of the sight ray, stops tracing the light path. Generates the light test ray from first visible point to the light source and generates intersection of sampled light rays.

3) According to the principle of path tracing, the light source is used to calculate the illumination for the visible point or the flux gathered from sampled intersection based ray tracing. After the paths interaction with the scene objects in various ways, their contribution is accumulated in the final results.

We divide the 3D into two categories at the beginning of modeling: static and dynamic geometric objects. Our test scene is divided as shown in figure 3.

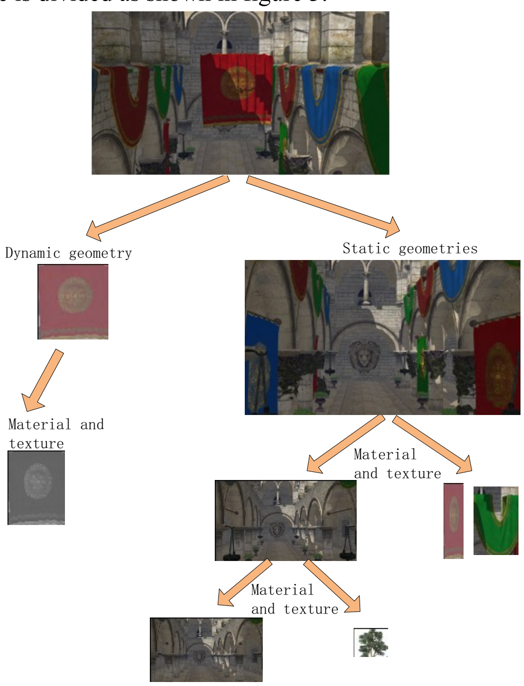

Figure 3. Partition of geometric object and scenario hierarchy structure.

D. Illumination of Path Tracing

In our method, only the closest intersection is tracked from view point to generate an eye sub-paths. The visibility test is started from the first intersection by random sample the light rays emitted from light source on the basis of a reversed ray tracing method, which determines the intersection with the scene surface according to the surface BRDF and track recursively. The reflection, and refraction is shown in figure 4.

If the rays between the intersected point of that come from the light source to the scene and the visibility point are not occluded by other geometries, the irradiance of the point is calculated and the light transport path of the point is recorded in a PATH structure. The probability of connections for our method is approximated based on the random sampling as our main goal is to reduce the computational cost of movie production with customer level graphic cards. The practical results show that this approach works well and satisfies to the requirement of our applications.

The light transport path is stored in the way of recording the each endpoint of the light. For this purpose, we design a PATH data structure that contains the pointer variables P_1 and P_2 to point to the list type. Pointer P_1 points to path list of sampling light with respect to each light source. Pointer P_2 points to the intersections of corresponding sampling rays with dynamic objects and static objects in terms of the vertex coordinates in SSG and DSG lists.

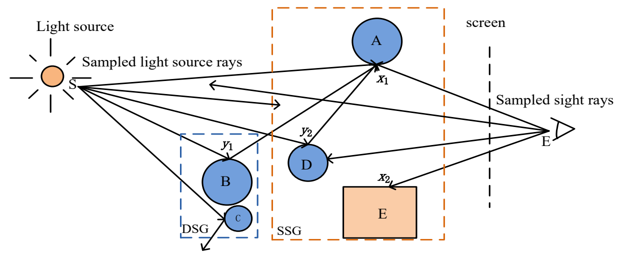

Figure 4. Schematic representation of path tracing in our method.

Let point x_1 is the first intersection of the light with the scene surface and x_1 belongs to static geometric object. Therefore, the coordinates of the point x_1 are stored in the vertex list pointed by pointer P_2 . On the other side, the sampled rays of light source is identified if it has contribution to point x_1 and it will also be stored in SSG table. Figure5 shows the storage structure and process of a frame. If the static geometry is occluded by the dynamic object, the associated transport path needs to be recalculated and stored in the SSG in the next frame.

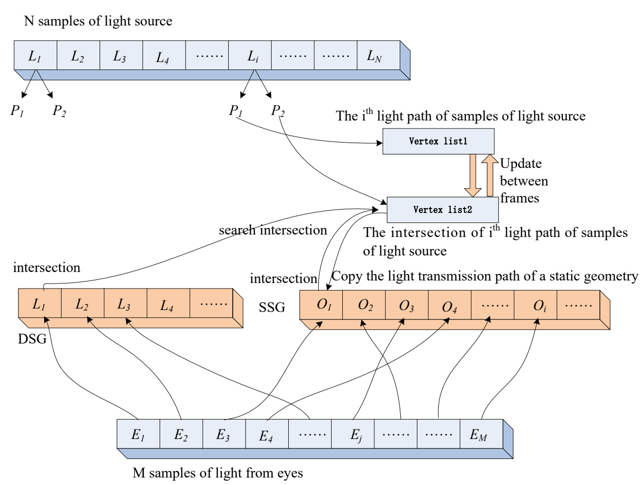

Figure 5. SSG organization and management.

III. Experiments and Analysis

The experimental environment of the algorithm was the Intel (R) Xeon (R) CPU, E3-1225 V3 3.20GHz together with 8 GB @ RAM and NVidia Quadro K5000 graphics card, running environment for accelerated ray tracing GPU NVidia 's OptiX SDK 3.7.0. BVH was used to accelerate the interaction provided by OptiX. Figure 6 showed the comparison of rendering effect with the standard ray tracing algorithm and the proposed method with SSG management respectively. Due to the insufficient light sampling, temporal flicker raises in figure6 (left). Our method reused part of the static scene light transport path and the flicker was significantly reduced at the same computing time. Moreover, increasing the number of sampling test rays can further improve the rendering effect, and reduce the flicker, as shown in figure 6 (right) as well as in figure 7 (left), where the number of sampling rays per pixel was 1.5. Figure 7 (right) shows the effect of 2.5 samples per pixel, which obtained precisely effect with lower running time compared the tradition method. However, this will increase the computing time and cannot achieve the interactive frame rate. Figure 8 showed the rendering effect using our approach, which got plausible effect in dynamic scene. We compared the traditional Monte Carlo Path tracing and our method using an area light source. Noise was reduced by reusing the sampled rays which increased the utilization of the lights that have the contribution to the scene.

Figure 6. Comparison of traditional ray tracing and our method. Traditional ray tracing (left); our method result with 1.5 samples per pixel (right).

Figure 7. Comparison of our method with different samples. 1.5 samples per pixel (left) and 2.5 samples per pixel (right).

Figure 8. Comparison of traditional Monte Carlo Path tracing and our method, path tracing (left), our method (right) at the same resolution of sampling. The number of paths is 10 per pixel.

Finally, we tested the running time of dynamic scene with 800 frames of our method and ray tracing algorithm. The result (in figure 9) shows that about 30% calculation of the algorithm is reduced overall compared to traditional ray tracing by utilizing our method.

IV. Conclusions

In this paper, we presented a fast method to generate the global illumination by a variation of bidirectional path tracing method. The geometric objects in the 3D scene were divided into static geometries and dynamic geometries. On the basic of random sampling, the light rays from the light source were reduced by reusing the light transport of static geometries. SSG and DSG structures were built to organize and manage the scene by storing the relative test rays. Experiment result shows that our method improved the computing performance effectively and reduced the flicker and noise issues of dynamic scene due to the insufficient samples.

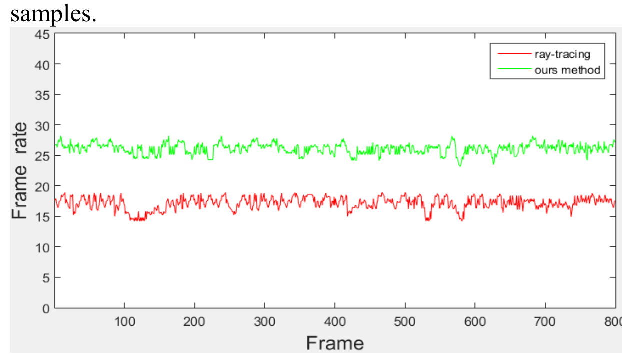

Figure 9. The comparison of running time of traditional ray tracing and ours.

Acknowledgment

This work was financially supported by Natural Science Foundation of Jilin of China (20170101005JC), Jilin province industrial innovation special funds projects (2016C091), Open platform project of Jilin Natural Science Foundation (20150101015JC) and Jilin Province Key Project of Science and Technology (20160204019GX).

REFERENCES

[1] Wald I, Benthin C, Slusallek P, “A simple and practical method for interactive ray tracing of dynamic scenes,” Submitted for publication, also available as a technical report at http://graphics. cs. uni-sb. de/Publications, 2002, 6: 6.

[2] Jim Kajiya, “The rendering equation,” Computer Graphics Proceedings of SIGGRAPH, 1986, 20(4) pp. 143–150, doi>10.1145/280811.280987.

[3] Georgiev I, Krivánek J, Davidovic T, et al., “Light transport simulation with vertex connection and merging,” ACM Trans. Graph., 2012, 31(6), 192, doi:10.1145/2366145.2366211.

[4] Hachisuka T, Pantaleoni J, Jensen H W., “A path space extension for robust light transport simulation, ”ACM Transactions on Graphics (TOG), 2012, 31(6), 191, doi:10.1145/2366145.2366210.

[5] Šik M, Otsu H, Hachisuka T, et al., “Robust light transport simulation via metropolised bidirectional estimators, ”ACM Transactions on Graphics (TOG), 2016, 35(6), 245, doi:10.1145/2980179.2982411.

[6] Anthony P, Loïc B, Mathias P, et al., “Combinatorial Bidirectional Path-Tracing for Efficient Hybrid CPU/GPU Rendering, ”Computer Graphics Forum. Blackwell Publishing Ltd, 2011, pp.315-324, doi: 10.1111/j.1467-8659.2011.01863.x.

[7] Popov S, Ramamoorthi R, Durand F, et al., “Probabilistic Connections for Bidirectional Path Tracing,” Business Process Management Journal, 2015,34(4):75-86, DOI: 10.1111/cgf.12680.

[8] Petrik C, Tomas A., “Practical Product Importance Sampling for Direct Illumination., ”Computer Graphics Forum, 2008, 27(2), pp. 681-690.

[9] Walter B., Khungurn P., Bala K., “Bidirectional lightcuts, ”ACM Transactions on Graphics (TOG), 2012, 31, 4, doi:10.1145/2185520.2185555

[10] Vorba J, Karlík O, Šik M, et al., “On-line learning of parametric mixture models for light transport simulation,” ACM Transactions on Graphics (TOG), 2014, 33(4): 101, doi:10.1145/2601097.2601203