International Journal of Engineering & Technology, 7 (3.24) (2018) 658-661

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Non-Voxel-based Interactive Indirect Lighting Using Layered Reflective Shadow Maps

[1]Bo Zhang, [2]JinHyung Choi, [*3]KyoungSu Oh, [4]HyungIlChoi

[1,2,3,4]Dept. of Digital Media, Soongsil University,378 Sangdo-ro Seoul, KS013, South Korea
**Corresponding author Email: oks@ssu.ac.kr*

Abstract

Background/Objectives: In recent years, global illumination (GI) has become a significant research area in computer graphics. The game industry has a lot of market demand for real-time GI rendering.
Methods/Statistical analysis: The key idea of our approach is generating G-buffered layered depth, normal and flux maps (LRSMs) from the light source in the first rendering pass. Subsequently, we sample LRSMs which are stored in GPU memory and combine normal distribution approximation method for deferred shading in the second rendering pass to simulate indirect light.
Findings: We present a non-voxel-based interactive indirect illumination algorithm which can approximately calculate the intensity of one-bounce indirect light in traditional graphics pipeline. Without voxelization, we are able to avoid using the complex octree data structure of voxels, which can improve the rendering performance by simplifying the process while obtaining similar results of voxel cone tracing. The test rendering rate in sponza scene reaches average 69.2fps.
Improvements/Applications: Our algorithm also supports diffuse and glossy reflection scenes and the objects and light sources are able to dynamically move at a given scene.

Keywords: *global illumination, layered reflective shadow map, non-voxel-based, cone tracing, normal distribution approximation*

1. Introduction

How to estimate indirect light is an essential research area in computer graphics, but its calculation is usually expensive and challenging. It cannot be high-quality and real time performed with recent hardware due to dealing with the integral computation of the rendering equation [1]. This equation is a kind of integral formula for gathering the intensity of lighting around the hemisphere of a point, and modern hardware cannot efficiently support computing massive integral computation. Usually, a random sampling method called Monte Carlo is needed to simulate integral result but a lot of sample numbers should be launched for good visibility. Therefore, traditional GI approaches cannot be run in real time although they can perform excellent rendering results, such as radiosity, ray tracing, path tracing, photon mapping, and so on. However, it also usually faces substantial and complicated computing process on hardware for better visual effects. Several GI methods have been proposed to solve indirect illumination problem in real time, such as reflective shadow map, light probes, voxel cone tracing, and so on. Recently, real time calculated global illumination is becoming more and more significant in the modern computer games area [2]. Many researchers are dedicated to studying and solving the problems of rendering indirect lighting because it can greatly improve the quality of the rendering result in games. The most famous GI method is PRT [3] which precomputes the radiance transfer situations and stores the environment light information with spherical harmonics around each vertex. Another novel GI algorithm is sampling light probes [4], a kind of small environment light ball, placed on an empty space to approximate the indirect light. To face these challenges, "reflective shadow map" and "voxel cone tracing" were combined in this paper while voxels and octree structure were avoided for a more simple computation and better performance of indirect light. A similar approach with reflective shadow map was introduced to approximate indirect light. In the first pass, layered normal, in-depth, and flux maps were created from a per-light source of a whole given scene and stored in GPU memory synchronously for the resources of the next rendering pass. At the second pass, a final gathering was made to calculate indirect light via non-voxel-based cone tracing. Finally, indirect light result (from the second pass) was additionally blurred for obtaining better visual lighting effects with less artifacts.

2. Related Work

In the real time indirect illumination field, there are several common methods to approximate the one-bounce indirect light's intensity. The first method is a reflective shadow map-based global illumination algorithm, and the second method is a voxel-based global illumination algorithm. In this paper, these two algorithms became the focus, and their quintessence was absorbed to build a new method that will simplify the GI calculation.

Virtual point lights algorithm represents an effective approach [5] for indirect light performance but could suffer bright spots problem on a sampling process. Reflective shadow map (RSM) [6] introduced a novel interactive indirect light simulation approach with traditional graphics pipeline, but the result cannot avoid some unfriendly color bleedings and artifacts. When two or more objects appear in a scene, a phenomenon may occur where indirect light can penetrate an object's body and thus ignore the occlusion information.

Another famous approach for calculating indirect light is voxel cone tracing [7]. In this paper, the author voxelizes the whole scene via GPU geometry shader, then generates octree structure for quick cone tracing the indirect light. It has a good one-bounce result in real time; however, ample memory space is needed when the scene becomes complex; fast voxelization and octree building are also extremely difficult in GPU.

Recently, a paper has tried to combine “layered reflective shadow maps” and “voxel cone tracing” and these two different ideas attempted to solve the GI problem [8]. It stores some layered, in-depth, normal, flux maps in G-buffer serving as resources in the first rendering pass. At the next pass, deferred rendering techniques calculate the part of indirect light by sampling LRSMs. However, it also has the same disadvantages with voxel cone tracing; that is building a complex octree structure and taking voxelization for the whole scene.

Additionally, many researchers try, via artificial intelligence, to address global illumination issues in real time. Radiance regression function [9] method uses a simple neural network—a total of four layers—to train the R, G, and B channels’ color intensity. It can obtain fine GI performance but needs too much processing and work for preparing massive input data and a lot of time for training.

In contrast, this study’s approach is similar to the paper [8] where LRSMs were used but this study is committed to not using complex octree structure and voxelization processes for simplification. This method can show a similar GI rendering quality and real time frame rate with voxel cone tracing algorithm at a similar scene.

3. Overview

This paper’s approach is inspired by the reflective shadow map and voxel cone tracing algorithms. These steps were followed to achieve the said algorithms:

In the first rendering pass, LRSMs from a light source were generated and stored. According to the world space normal, 9-direction cone tracings were taken for a final gathering on the hemisphere of each position. For each direction, world space position was sampled in turnwithin the cone, and a sample range was decided for using mip-map depth buffer to approximate ambient average depth. Next, LRSMs were read at a specific sample position for getting normal, depth, and color information at that point. After which, depth normal distribution was performed to compute the percentage of indirect light by a method that is similar with the variance shadow map algorithm. Finally, all of the directions’ lighting intensities were averaged and Gaussian blur processing was taken to reduce some artifacts and for better visualization to be the final indirect light result.

The main approach pipeline of our paper is demonstrated as in [Figure 1].

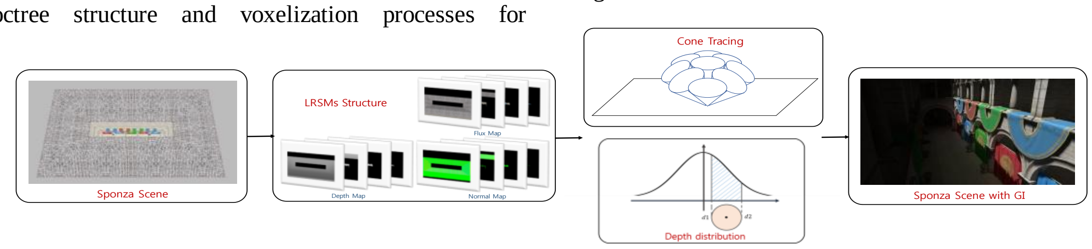

Figure 1: Pipeline

3.1 Layered Reflective Shadow Maps

Reflective shadow maps (RSMs) method for global illumination first appeared in a previous paper [6]. At that time, although the study still had some shortcomings, its greatest advantage was its ability to simulate the indirect lighting effect in real time. Because only one bounce was considered, the author tried to render the entire direct lighting scene at the light source and store some rendering images in GPU memory to be resources. This provides an efficient and convenient method for the indirect lighting calculation at the next pass, and the GPU shader can store and sample RSMs fast. This is why the skill to solve GI problems was chosen. [Figure 2] shows the examples of LRSMs rendered from the Sponza scene.

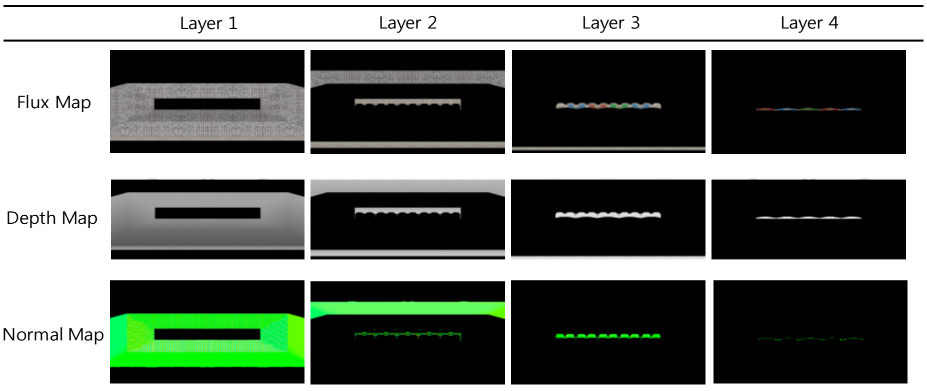

Figure 2: LRSMs of Sponza scene

Layered refers to a means of dividing an entire scene according to the depth values seen from the light source. Segmenting a scene artificially would improve the accuracy of the final estimate result because cone tracing was taken for the final gathering, which is an algorithm to track the light conditions around the target point. Often in complex large-scale scenes, the tracing distance that one needs is smaller than the size of the whole scene. It is particularly important to arrange the sampling RSMs resources reasonably. Therefore, a layered scene is taken to render the direct light scene and to obtain flux map which helps in sampling accuracy from the RSMs around the target point. Additionally, one does not evenly separate scenes but according to the shape of the scene. Where objects are more concentrated, they are more stratified. For example, in the Cornell-box scene, one can separate it with a total of five layers as in [Figure 3].

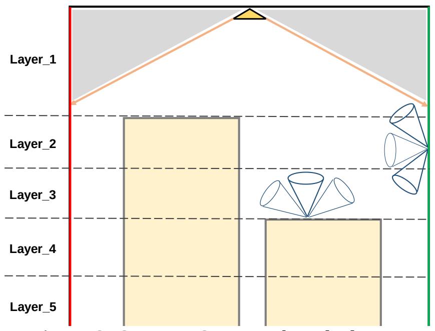

Figure 3: Segment Scene with multi-layer

3.2 Cone Tracing for Final Gathering

The cone tracing method is described in detail in the paper [7], and it is mainly for the final gathering to collect the one-bounce reflected light surrounding a position. One uses a similar approach but the sampling target resources are different. In voxel cone tracing, light information is gathered from voxel structures but light information is gathered from LRSMs.

3.2.1 Radiance Function

In a scene with a single light source, the final color intensity of a vertex is itself emissive radiance adding reflected radiance that is from its ambient environment. This study merely concentrates on how to estimate the part of the reflected radiance. According to one paper [1], one can calculate the reflected radiance L with an integral equation ①, x_p is a vertex position (target point), ρ is the BRDF function with viewing direction v at x_p, n is the world space normal at x_p, and $l_i(x_p, v_i)$ means the intensity of incoming radiance from the direction v_i.

As shown in equation ②, the reflected radiance$L$ can be divided into two parts: a local illumination (direct light) component L^0 and indirect illumination component L^+. This is similar with the ideaof Phong lighting model. L^0 can be calculated easily with an existing approach [10].

$$L(x_p) = \int \rho(x_p, v, v_i)(n, v_i) l_i(x_p, v_i) dv_i \quad (1)$$

$$= L^0(x_p) + L^+(x_p) \quad (2)$$

In this paper, taking i directions cone tracing for final gathering indirect illumination in each vertex is decided. For quality of $L^+(x_p)$, one suggests the maximum value N of i taking 9, including normaldirection, witheachvertices. Through a number of attempts, this is a balance value between rendering quality and running speed.

In each tracing direction, this paper calculates$i$-th $L^+(x_p)$value according to formula ③. $W \in [0,1]$ is a linear weight value calculated from the distance between x_p and current tracing sample positions. $G(d_1, d_2)$ is a percent value that is computed by a method called depth distribution approximation. For a detailed description one may refer to section 3.2.2. (n, v_i) means the $\cos(\theta)$ value, and the θ is the angle between the world space normal of x_p and current tracing direction vector. $C(r, g, b, r)$ is the mip-map RGBA color that is from current sample's LRSMs. Here, the mip-map level is decided by $\log_2(s_\varphi)$ and φ is the diameter size of s. Finally, the average results are from N-direction cone tracing to obtain $L^+(x_p)$.

$$L^+(x_p) \approx \frac{1}{N}\sum_{i=0}^{N} W(dist) G(d_1,\ d_2)(n, v_i) C(r, g, b, r) \quad (3)$$

Additionally, if there are some sharp artifacts in some places in the final indirect result, one could choose to use the appropriate Gaussian filter to blur the results in the final rendering pass to achieve better visualization.

3.2.2 Depth Distribution Approximation

In paper [11], the author estimates the intensity of the shadow by expected value $E(x)$, and in one paper [8] the author uses Gaussian distribution to approximate the percentage of indirect light's intensity. Normal distribution is a general method for calculating the probability problem. This probability would be helpful for one to decide the current mip-map value.

$$G(d_1, d_2) = G(d_2) - G(d_1) = \frac{1}{\sigma\sqrt{2\pi}}\left(e^{-\frac{(d_2-\mu)^2}{2\sigma^2}} - e^{-\frac{(d_1-\mu)^2}{2\sigma^2}}\right) \quad (4)$$

$$\mu = depth,\ \ \sigma^2 = E(depth^2) - E(depth)^2 \quad (5)$$

Thus, in this paper, one combines ideas to present a similar method named depth distribution approximation. At the first rendering pass, one also synchronously stores the traditional shadow map $(depth)$ and its squared value $(depth^2)$ on GPU buffer. Next, formulae ④ and ⑤are combined to calculate the μ and σ^2 values for obtaining the specific normal distribution function. According to this distribution graph, one can easily calculate the percent value $G(d_1, d_2)$. As seen in [Figure 4], d_1 and d_2 are the minimum and maximum depth values of current tracing sample range.

Pseudo code [Figure 5] would briefly describe the entire computation process. First, 9-direction cone tracing at each vertex should be taken. $d$means the current tracing distance from the vertex, and p is the percent value of how much color should be absorbed in this sample point. For each direction, indirect light L^+ should be calculated by formula ③while taking a loop. Finally, all N directions should be repeated while tracing with the process above until the tracing distance and the indirect color's alpha channel reach the threshold which was set by this study.

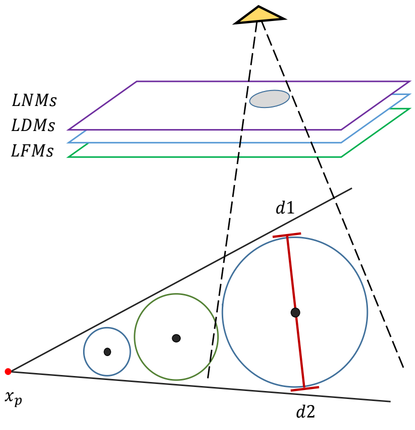

Figure 4: Cone Tracing with LRSMs

```
Algorithm                                  Approximate indirect radiance
$L^+ = 0$                                  • Outgoing radiance
for all directions do
  $d = 0$
  $p = 0$
  while $d < d_{max} \cap p < 1 \cap L^+_\alpha < 1$ do
    if current sample position in layer $i$ then
      if $-1 \le \cos\theta < 0$ then
        $L^+ \mathrel{+}= WGC_{layer\_i} \cos\theta$
        $percent \mathrel{+}= G$
      end
    end
    $d \mathrel{+}= 2U$
  end
end
```

Figure 5: Pseudo Code

4. Results and Discussion

In [Figure 6], all of the test images were rendered at 1280×720. The figure demonstrates the different rendering visual effects between only direct lighting and with indirect lighting in the same Sponza scene. The test environment of our algorithm was based on Core i5-6600 CPU, 16GB memory, and GTX980Ti GPU. The rendering was performed at average69.2fps with soft shadow in sponza.

Additionally, the problem of our method is on complex scene would appear striped artifacts. We suggest take blur processing to each frames at final rendering pass for improving artifact problem.

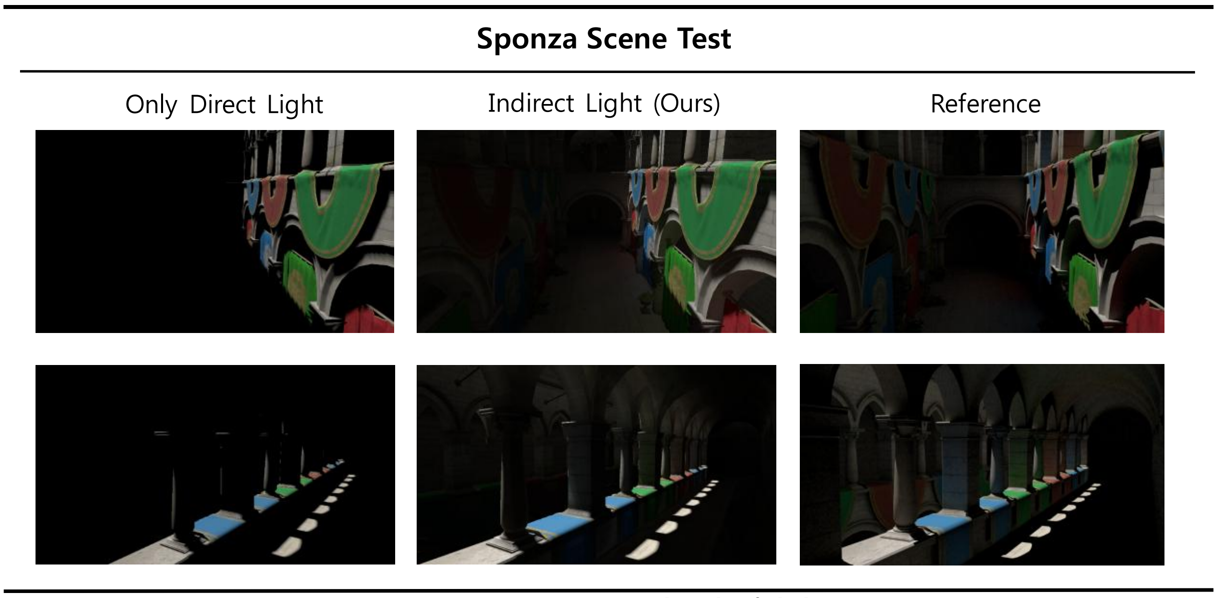

Figure6: Sponza scene Test Result (with soft shadow)

5. Conclusion

This paper described a non-voxel-based interactive algorithm to approximate the indirect lighting by using layered reflective shadow maps. The experiments demonstrate that taking a 9-direction cone tracing without building octree and voxelization can also get good GI visibility. The performance of the Sponza scene with the test hardware can reach an average of 69.2fps or more for real time rendering. Compared with the traditional algorithms, because the complex octree structure and voxelization are not used, the entire programming process is very simple and straightforward. The test result indicates that this study's algorithm looks well on simulating indirect light effect. Although there are some improvements, it may also be a good idea for approximate the indirect illumination.

Acknowledgment

This work was supported by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No.20170006810012002, Screen Archery Solution based on Hybrid Reality).

References

[1] Kajiya, J. T. (1986). The rendering equation. Acm Computer Graphics, 20(4), 143-150.
[2] Mittring, M..(2012).The Technology Behind the Unreal Engine 4 Elemental demo. SIGGRAPH 2012 Advances in Real-Time Rendering in 3D Graphics and Games Course, Retrieved fromhttps://de45xmedrsdbp.cloudfront.net/Resources/files/The_Technology_Behind_the_Elemental_Demo_16x9-1248544805.pdf.
[3] Sloan, P. P., Kautz, J., & Snyder, J. (2002). Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. Conference on Computer Graphics & Interactive Techniques, Vol.21, 527-536.
[4] Mcguire, M., Mara, M., Nowrouzezahrai, D., & Luebke, D. (2017). Real-time global illumination using precomputed light field probes. The ACM SIGGRAPH Symposium, 1-11.
[5] Keller, & Alexander. (1997). Instant radiosity. Proc of Acm Siggraph, 49-56.
[6] Dachsbacher, C., & Stamminger, M. (2005). Reflective shadow maps. Symposium on Interactive 3d Graphics and Games, 203-231.
[7] Crassin, C., Neyret, F., Sainz, M., Green, S., & Eisemann, E. (2011). Interactive Indirect Illumination Using Voxel Cone Tracing. Symposium on Interactive 3D Graphics and Games, Vol.30, 207-207.
[8] Sugihara, M., Rauwendaal, R., & Salvi, M. (2014). Layered reflective shadow maps for voxel-based indirect illumination. High PERFORMANCE Graphics, 117-125.
[9] Ren, P., Wang, J., Gong, M., Lin, S., Tong, X., & Guo, B. (2013). Global illumination with radiance regression functions. Acm Transactions on Graphics, 32(4), 1-12.
[10] Donikian, M., Walter, B., Bala, K., Fernandez, S., & Greenberg, D. P. (2006). Accurate direct illumination using iterative adaptive sampling. IEEE Transactions on Visualization & Computer Graphics, 12(3), 353-364.
[11] Donnelly, William, Lauritzen, & Andrew. (2006). Variance shadow maps. Si3d '06 Proceedings of the Symposium on Interactived Graphics & Games Acm, 161-165.