TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC QUY NHƠN

Một kỹ thuật lấy mẫu phục vụ hiển thị ánh sáng

Lê Thị Kim Nga[1,*], Đoàn Thị Xương[2], Lương Thị Mộng Duyên[3]

[1]Viện Nghiên cứu ứng dụng khoa học và công nghệ, Trường Đại học Quy Nhơn, Việt Nam
[2]Thư viện, Trường Đại học Quy Nhơn, Việt Nam
[3]Khoa Khoa học máy tính và Công nghệ thông tin, Trường Đại học Quang Trung, Việt Nam

Ngày nhận bài: 27/11/2020; Ngày sửa bài: 26/02/2021;
Ngày nhận đăng: 27/02/2021; Ngày xuất bản: 28/06/2021

TÓM TẮT

Tạo ra được cảm giác chân thực cho các hình ảnh 3 chiều là chủ đề rất được quan tâm trong lĩnh vực đồ họa máy tính và thực tại ảo. Một trong những hướng tiếp cận phổ biến cho bài toán này là sử dụng dò tia ngẫu nhiên trên cơ sở phương trình tô bóng. Bài báo trình bày một kỹ thuật lấy mẫu phục vụ hiển thị ánh sáng trên cơ sở một số tùy chỉnh với tiếp cận dò tia ngẫu nhiên theo phương trình tô bóng.

Từ khóa: *Lấy mẫu, Monte Carlo light transport, thực tại ảo, dò tia, tô bóng.*

**Tác giả liên hệ chính.*
Email: kimnle@arist.edu.vn

https://doi.org/10.52111/qnjs.2021.15301

QUY NHON UNIVERSITY
JOURNAL OF SCIENCE

A sampling technique for light display

Le Thi Kim Nga[1,*], Doan Thi Xuong[2], Luong Thi Mong Duyen[3]

[1]Research Institute for Applied Science and Technology, Quy Nhon Universiy, Vietnam
[2]Library, Quy Nhon University, Vietnam
[3]Faculty of Computer Science and Information Technology, Quang Trung University, Vietnam

Received: 27/11/2020; Revised: 26/02/2021;
Accepted: 27/02/2021; Published: 28/06/2021

ABSTRACT

Creating realistic sense for 3D images is a great topic of interest in the field of computer graphics and virtual reality. One of the common approaches to this problem is to use random ray tracing based on shading equations. The paper presents a sampling technique for light display based on a number of customizations with random ray tracing approach according to the shading equation.

Keywords: *Sampling, Monte Carlo light transport, virtual reality, ray tracing, shading.*

1. INTRODUCTION

Nowadays, virtual reality is one of the key science and technology fields thanks to its broad applicability in various aspects of life, such as health, education, architecture, military, tourism, and entertainment, etc. With many products being implemented, according to[1] the market for virtual reality products is predicted to grow from US $7.9 billion in 2018 to US $44.7 billion in 2024, with the annual growth rate of 33.5% during the forecast period.

Figure 1. A number of virtual reality applications

One of the most interesting research issues of virtual reality is the shading techniques for displaying high-fidelity 3D images. We have known that to see an object, the light from that object must reach our eyes. Light can come from a light source, hit the object and bounce off our eyes. It is also possible that light collided with another object or other objects before it touched the object. Hence when looking at a point on an object, there are countless rays of light that collide violently in the scene before hitting that point and reaching our eyes. Typically, studies will attempt to analyze and simulate part of that process to build realistic lighting display techniques.

One of the well-known studies on understanding how light travels through space to calculate emission levels at each point is the shading equation given by Kajiya.[2] Accordingly, the emission at a point is calculated internally by summing up the emission level itself and the intensity of the reflected radiation when the surface is illuminated by light sources.

**Corresponding author.*
Email: kimnle@arist.edu.vn

https://doi.org/10.52111/qnjs.2021.15301

$$L_0(x, \vec{\omega}_0) = L_e(x, \vec{\omega}_0) + \int_{\Omega(\vec{n})} f_r(x, \vec{\omega}_i \to \vec{\omega}_0).L_i(x, \vec{\omega}_i).\cos(\vec{n}_i, \vec{\omega}_i)d\omega_i$$

In which, L0 is the radiation intensity at x point in one direction $\vec{\omega}_0$. L0 will be calculated by the intensity of radiation itself. Le and the sum of reflections is calculated using a distribution function of two-dimensional reflection.[3] If the surface is not a source, value Le is equal to 0.

According to the shading equation, reflected radiation is synthesized in all possible directions of the incident light rays. This calculation is impractical, so we can apply Monte Carlo numerical simulations to sample the incident light rays, thereby turning the integral into a finite sum.

A study using light-based simulation with this approach is Cinematic Rendering,[4] which results in high-quality 3D images from tomography data. The technique is built on the basis of applying Monte Carlo method to randomly sample the traced rays along with using shading equation on cube data.

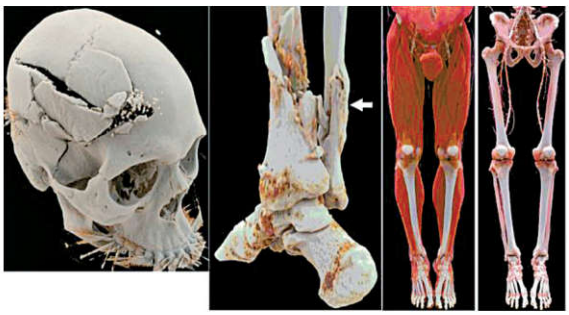

Figure 2. A number of images of Cinematic Rendering[4]

Using ray tracing to approximate the actual transmission of light by the Monte Carlo method is also one of the common approaches in computer graphics to create images for high-quality 3D animation. This access is used in some Pixar animation studio data. By rendering high-definition animations, the amount of rays used is extremely large, which also leads to the need for computational time. According to Christensen team,[5] it takes about 15 to 24 hours for a frame rendering for films like Cars and Monsters University.

The paper presents a sampling technique for displaying light according to the random ray tracing approach to perform light synthesis based on shading equations. Accordingly, Section II will detail the theoretical contents of techniques with reflection ray sampling, light source point sampling as well as random ray interruption. In Section III, there are some test results with analysis of the output quality as well as the execution time corresponding to the technical customizations. Conclusion is at the end.

2. A SAMPLING TECHNIQUE FOR LIGHT DISPLAY

2.1. Directional scattering

When light strikes a surface, what is visible to the surface is reflected rays. Figuring out how to estimate the direction of the reflected ray is an important criterion in simulating light hitting that surface. For some surfaces, such as mirrors, the reflected light is considered to be perfect according to the laws of light reflection, ie the incident ray and the reflected ray align with the surface normal at equal angles. With some non-glossy surfaces, the light will scatter depending on the extent. And, for example, with the fabric surface, it can be considered that light is scattered evenly in all directions of the sphere half considered in the distribution function of two-dimensional reflection.

There are many ways to estimate the direction of the reflected ray. In this paper, we test reflected light rays at an intermediate level between uniform scattering and mirror reflection. Accordingly, the reflected light ray will have the taken direction as the directional ray generated according to standard distribution around the position of the reflected mirror ray.

Mirror reflection is referred as ds, ds standardized to unit vector. We set up a space

with the coordinate origin at the reflection point and there are 3 base vectors, u, v and ds.

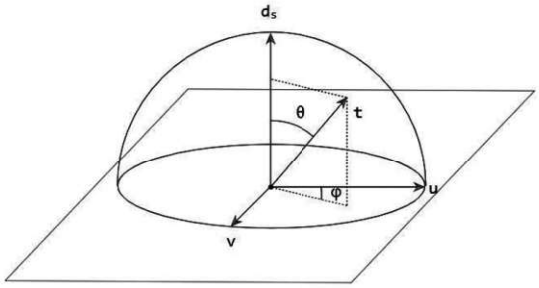

Figure 3. Space for generating reflection rays

In Figure 3, we can see space for generating refection rays. In this space, t is the generated reflected ray; t originates from the reflection point and has a length of 1. Reflective generation is based on two angles θ and φ. In which, φ is randomly distributed evenly within [0, 2π], θ is generated as a standard distribution with an expectation of 0 then takes the absolute value. θ is also limited to upper bound as $\frac{\pi}{2}$.

2.2. Sampling the light source point

While performing the sampling of the traced rays to perform Monte Carlo simulation for the shading equation, the path of the sampling ray needs to collide with the light source for that sampling to be meaningful. That is when the radiation value at the new observation point is updated by the sampling ray. With extremely large light sources, such as the sky, most sampling rays make sense. However, with small light sources, such as light bulbs in the room, or the sun at extremely remote locations, it is clear that the probability of the sampled rays hitting the light source is quite low. That is likely to waste a lot of computing resources.

Instead, based on some suggestions from Shirley's work,[6] we performed reflected radiation at each location that would be sampled by a direct ray to each light source if it is not covered and another random ray in which the direct ray from the light source will be sampled based on a random sample of the light source. Specifically, in this case, we assume that the light source is a sphere and we randomly take points according to a uniform distribution on the light source sphere. This point together with the point to be calculated will be used to construct the ray directly with the light source.

We consider the point on the sphere described as Figure 4. In this, r is the sphere radius, and this value will be fixed for each specific sphere. The values of θ, φ will change when we consider different points on the sphere.

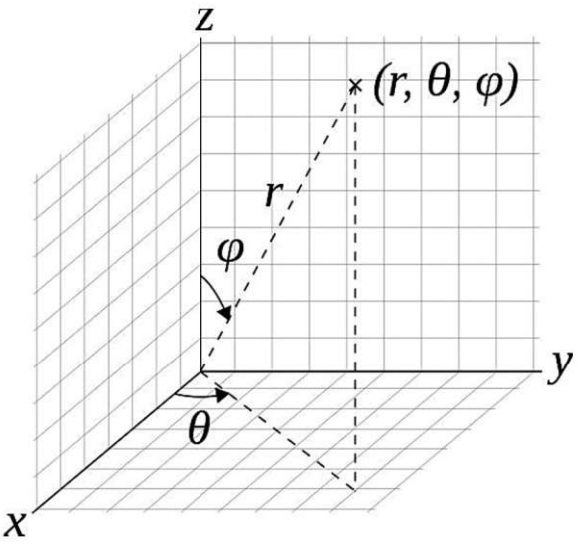

Figure 4. Points on the sphere represented by a set (r, θ, φ)

The generated score will have the following coordinates:

$$x = rsin\varphi cos\theta y = rsin\varphi sin\theta z = rcos\varphi$$

Based on this we have the first case that is randomly generated by uniformly distributed θ in [0, 2π] and φ in [0, π].

However, according to Simon,[7] the generation in the first case will cause the generated points to be concentrated at the poles, so an alternative case is case 2 with $\varphi = cos^{-1}(1 - 2u)$, where u is randomly distributed evenly in [0, 1] and θ is generated the same way as before.

In addition,[7] it is also described an alternative case which is case 3: independent generation 3 values of x, y, z randomly according to the normal distribution, calling $V = (x, y, z)$. At that point $P = \frac{V}{\|V\|}$ would be randomly distributed evenly across a sphere.

https://doi.org/10.52111/qnjs.2021.15301

2.3. Ray interruption

Sampling of rays was carried out on the basis of physical light. In practice, however, a ray can strike continuously until it runs out of energy. That means one ray may be sampled millions of times or more, which means it is not practical to perform calculations. Moreover, having collided too many times, its contribution to radiation has also become very small. To interrupt ray generation, we experimented with a reflectance threshold value $t \in [0,1]$. Thus, if $u > t$, the ray generation process will be interrupted. In which, u is taken randomly according to uniform distribution in [0,1]. The t threshold also corresponds to the fact that surfaces have high light absorption capacity, when the amount of reflected rays will be less than those with less light absorption. In addition, to limit the number of rays, we set a parameter of the ray interruption limit, that is, the maximum number of reflections possible of the rays in the rendered scene.

3. EXPERIMENT

The program was tested and installed on the basis of Visual C++ 2015, with the support of two open sources OpenCV[8] and montelight-cpp.[9] The rendering process is performed purely on the CPU. The test scene is designed based on the Cornell box model and conducted on an Intel Core i7-4510U 2.6GHz computer.

Figure 5. Rendering images from left to right of different cases of spherical reflection: uniform scattering, directional scattering, mirror reflection

We can easily see the difference between the effects in Figure 5. In detail, the use of directed scattering produces an intermediate result compared with uniform scattering and mirror reflection. This is quite understandable and has been predefined from the theoretical content. Next, we sample the light source point sample in 3 described cases in Section 2.2. To show how points are generated, we visualize the sampling points. In which, 800 points were sampled on a case by case basis described as Figure 6.

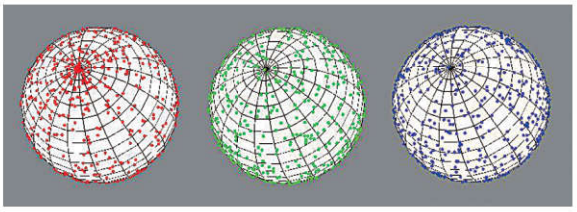

Figure 6. Description of sampling random points on a sphere in 3 cases: Case 1 in red, Case 2 in green, Case 3 in blue

Obviously, the points focus more on the pole in Case 1 when we randomly generate both angles in a uniform distribution. In the other two cases, the points are evenly distributed on the surface of the sphere. To illustrate the effect of 3 light point sampling strategies, we perform image rendering with two cases where the light source is a large sphere and the light source is a small sphere. Result is described as Figure 7.

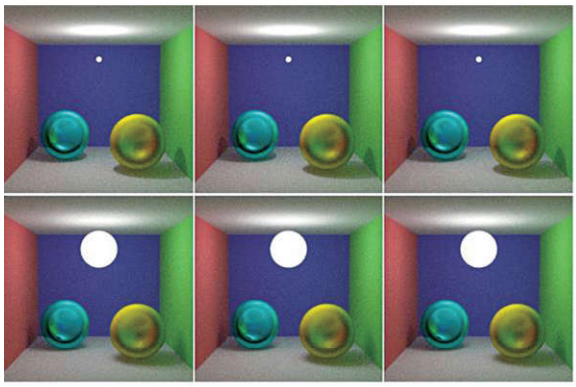

Figure 7. Rendering images according to 3 light source sampling points: cases in order of 1, 2, 3 turn from left to right and the top row is with a small light source, the bottom row is a large light source

In the test, with each rendering, 50 rays were sampled. It was found that there was no clear difference in the influence of 3 light source sampling cases for both small and large light sources. Thus, although Case 1 is not really random according to the uniform distribution of the sphere, we can use it to get the same result.

In order to experiment with the ray interruption, the threshold for generating reflected

https://doi.org/10.52111/qnjs.2021.15301

rays is set individually between different objects in the scene, while the ray interruption limit will be fixed in the whole scene. We also performed a run time comparison between ray break limits with 128x128 image rendering test and 50-ray sampling.

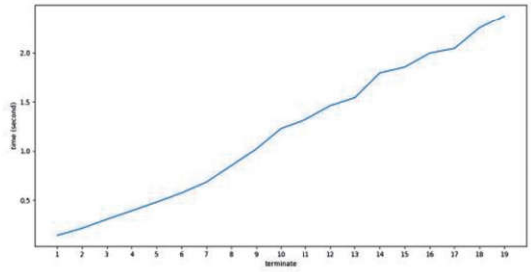

Figure 8. Diagram between ray interruption limit and runtime

We observe Figure 8 to visualize the relationship between the interrupt limit and run time. There is a relatively clear time difference between the ray interruption limits and the change is almost linear in proportion to the ray interruption limits. This is quite understandable because the calculated volume is equivalent at once taking rays at a point.

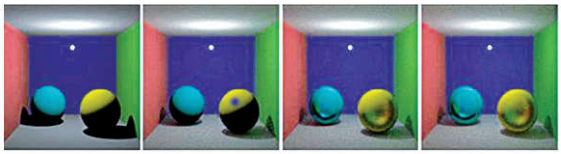

Figure 9. Images rendered with ray interruption limits, respectively 1, 2, 3, 4

To visualize more clearly the effect of ray interruption limits to rendered image, we observe Figure 9. This is image results shown corresponding to the different ray interruption limits. With Level 1, the shadow areas are very clear because the ray only reflects once, so in addition to direct light from the light source, there is no other source. Therefore, places hidden from the direct light source will have no light. In addition, with a limit of 2, the hidden regions have secondary reflecting light, which is no longer dark, but these must be higher interruptions, mirror reflection and directional scattering effects on surface of the two new spherical balls is clearly shown.

4. CONCLUSION

The paper presents a sampling technique for displaying light according to the random ray tracing approach to perform light synthesis based on shading equations. Research simulating light effects to express images with high fidelity is one of the most interesting research issues of virtual reality. This problem is even more meaningful in terms of the field of virtual reality, which becomes popular in the world in general and in Vietnam in particular. The paper presents a sampling technique for light display on a random ray based on shading equations. These are important initial research results for the team to serve as the foundation for further studies in the field of graphics and virtual reality.

REFERENCES

1. Markets and markets, Virtual Reality Market with COVID-19 Impact Analysis by Offering (Hardware and Software), Technology, Device Type (Head-Mounted Display, Gesture-Tracking Device), Application (Consumer, Commercial, Enterprise, Healthcare) and Geography - Global Forecast to 2025, <https://www.marketsandmarkets.com/Market-Reports/reality-applications-market-458.htm>, last accessed on 23/5/2020.

2. J., T. Kajiya. *The rendering equation,* ACM SIGGRAPH Computer Graphics, Proceedings of the 13th annual conference on Computer graphics and interactive techniques, 31/8/1986.

3. Forsyth, A . David. *Computer vision: A modern approach,* 2nd edition, Pearson, New York, 2012.

4. M. Eid, C. D. De Cecco, et al. Cinematic rendering in CT: a novel, lifelike 3D visualization technique, *American Journal of Roentgenology*, **2017**, *209*(2), 370-379.

5. P. H. Christensen, J. Fong, D. M. Laur, and D. Batali. Ray tracing for the movie 'Cars', *IEEE Symposium on Ray Tracing*, **2006**, 1-6.

https://doi.org/10.52111/qnjs.2021.15301

6. P. Shirley, C. Wang, and K. Zimmerman. Monte Carlo techniques for direct lighting calculations, *ACM Trans. Graph,* **1996**, *15*(1), 1-36.

7. C. Simon. Generating uniformly distributed numbers on a sphere, <http://corysimon.github.io/articles/uniformdistn-on-sphere/>, last accessed on 23/5/2020.

8. Open source computer vision library, <https://sourceforge.net/projects/opencvlibrary/>, last accessed on 23/5/2020.

9. Montelight, <https://github.com/Smerity/montelight-cpp>, last accessed on 23/5/2020.