DOI: 10.15276/hait.01.2021.4
UDC 004.925.8

METHOD FOR CALCULATING THE REFLECTION FUNCTION OF GLOBAL ILLUMINATION WITH PERTURBATION FUNCTIONS

Roman Y. Chekhmestruk[1)]
ORCID: 0000-0002-5362-8796; Rc.ua@3dgeneration.com
Pavlo I. Mykhaylov[2)]
ORCID: 0000-0001-5861-5970; pm@3dgeneration.com
Sergey I. Vyatkin[3)]
ORCID: 0000-0002-1591-3588; sivser@mail.ru
[1)] 3D GENERATION UA, 37, Pirogova Str. Vinnitsa, 21021, Ukraine
[2)] CEO 3D GNERATION GmbH. Märkische Straße 90-92, Dortmund, 44137, Germany
[3)] Institute of Automation and Electrometry, SB RAS. 1, Ak. Koptyuga, Novosibirsk, 630090, Russia

ABSTRACT

The advent of new hardware and the ever-increasing demands on the complexity of scenes are forcing the development of new approaches for calculating lighting. Modern visualization requires not only photorealistic, but also physically correct calculation of lighting. The core of any algorithm for calculating global illumination is the calculation of the illumination integral over the hemisphere. The aim of the work is to develop an effective visualization method based on the radiance caching and reprojection. This paper presents a modified method that eliminates the shortcomings of the reprojection algorithm for the radiation cache. Reprojection is not a fast procedure, since it is necessary to normalize the vector and calculate the inverse trigonometric functions if spherical coordinates are used to parameterize the hemisphere. In addition, it is necessary to use the z-buffer and solve the problem with the voids that will remain after the projection. In addition, for the calculation of illumination from extended sources, the known algorithms have certain disadvantages and are designed for a very limited number of cases. Therefore, in this paper, a universal algorithm is developed for calculating scenes of great complexity that have extended light sources, as well as secondary sources. The difficulty lies in the fact that the same point of the surface can be completely in the shadow or completely in the light from some light sources (the rays to such sources are coherent) and is in the penumbra from other sources (where the coherence of the rays is small). Therefore, simple methods of interpolation or extrapolation of lighting is not suitable. Additional difficulties arise with secondary light sources, which are implicitly represented in the scene and their location is not known in advance. The proposed method caches the incident radiation function and uses the calculated values at adjacent surface points, which significantly reduces the number of ray traces and the calculation of the reflection function. Unlike other radiation caching algorithms, the proposed method can work with high-frequency data. In comparison with the classical implementation of the Monte Carlo method, the method gives an acceleration of an order of magnitude with comparable calculation accuracy. The method can be used to calculate the final collection in the methods of photon maps and emissivity, illumination from an environment map set with a large dynamic range, shadows from large area light sources, "blurred" reflections, etc.

Keywords: Perturbation functions; geometric objects; global illumination; radiance; reflections; photon maps; ray tracing

For citation**: Chekhmestruk R. Y., Mykhaylov P. I., Vyatkin S. I. Method for Calculating the Reflection Function of Global Illumination with Perturbation Functions. *Herald of Advanced Information Technology*. 2021; Vol.4 No.1: 47–56. DOI: 10.15276/hait.01.2021.4**

1. INTRODUCTION

Nowadays computer graphics oriented to 3-D scene rendering has attained significant success.

The global lighting system with hardware support allows you to get realistic images in real time. Along with the global lighting system, materials based on the physical model further enhance realism. In the global lighting system, ambient lighting simulation and reflection play an important role. To do this, special markers (probes) are placed in the right places in the scene, which preserve the reflections around them in cubic textures. When moving between markers, the most significant pair is selected, and the reflections received from both are mixed. Reflections can be calculated in real time, at the time of marker activation, or even controlled by a script, where you can implement, for example, timers. Such systems are often implemented in games where reflections are constantly needed, in particular, to simulate the paint material. Simulating secondary lighting also greatly increases the realism of rendering. In the real world, many materials re-reflect the light that falls on them and become light sources themselves. To calculate such indirect lighting in real time, modern computing power is not enough, but it can be pre-calculated for static objects and sometimes recalculated for dynamic ones. The calculated data is packed into textures, which are then used in real-time rendering. These textures are called light maps.

For example, several mechanisms affect the formation of global lighting. Assigning a light

© Chekhmestruk R. Y., Mykhaylov P. I.,
Vyatkin S. I., 2021

source of the baked or mixed type. Such a light source will work through a light map and not affect dynamic objects (the baked type) or work for dynamic objects as a full-fledged light source (the mixed type). Illumination markers are a three-dimensional graph whose nodes store the level of illumination created by different light sources. In real-time rendering, data interpolated from the graph grid is used to calculate lighting. When calculating indirect illumination, we can assume that all surfaces are perfectly flat, i.e. they re-reflect light in the same way in all directions, and we can take into account the predominant direction of re-reflection using data from the normal map. This is the essence of this technology. The directional seculars mode is also supported, which allows you to take into account glare, which allows you to create a full-fledged secondary lighting. All this has many parameters that allow you to take into account the balance of performance-quality.

2. PREVIOUS WORK

Consider the existing algorithms for calculating illumination from extended light sources. Using ray tracing and Monte Carlo methods (for example, path tracing), it is theoretically possible to calculate lighting for any scene, within the framework of the visualization equation. However, even for simple scenes, calculations can take hours, days, and weeks. To speed up calculations, many methods of reducing variance were proposed, for example, significance sampling and adaptive sampling, which only partially solved the problem. For sampling by significance, it is necessary to know at least the approximation of the integrand and be able to reverse the distribution function, which is difficult to do for any complex scenes. Adaptive methods need to have an initial selection.

To apply the emissivity method, it is necessary to make two assumptions about the three-dimensional representation of the scene: all light sources are considered as diffuse emitters, and all surfaces are considered as diffuse reflectors. These two assumptions lead to the fact that the outgoing radiation becomes independent of the direction, since the Bidirectional reflectance distribution function (BRDF) becomes a constant for each point of the surface. In this way, the integration problem is transformed into a finite element problem. For the algorithm to work correctly, it is necessary that the scene is divided into elements according to the lighting (which is unknown), so the method is not well suited for complex scenes.

To calculate global illumination, emissivity algorithms are used [1-4]. A small number of virtual point light sources are generated using the quasi-random walk method. Then the scene is rendered using shading from these virtual sources, as from ordinary ones. Most calculations can be performed on the GPU. This algorithm uses discretization of secondary light sources only, so it has no artifacts related to geometry discretization, but creates its own artifacts due to discretization of virtual sources, which manifests as a set of hard shadows where there should be a soft shadow. In progress [5, 6] interactive ray tracing of fast global illumination is presented, where ray tracing with intermittent sampling of virtual light sources performed shading. Fast emissivity and fast global lighting is bad for non-Lambert BRDF and scenes with lots of obstacles.

The classical shadow volume algorithm for hardbound shadows has been generalized to soft shadows [7]. Since the algorithm is based on finding the silhouette of the object that casts the shadow, the speed of soft shadow volumes depends on the number of edges that create the shadow. Therefore, this method is poorly suited for complex scenes with large extended light sources.

Methods that are based on shadow maps [8] are better scaled under the size and complexity of the scenes compared to shadow volumes. The scene is rasterized from the light source, and the pixel depth is written to the buffer. Further, when rasterizing from the camera, you can use the z test to determine whether a surface point lies in the shadow of a given light source or not. An important role is played by hardware support for rasterization, which is available in modern GPUs.

Early work on soft shadow maps simply averaged a few hard shadows, which naturally severely limited the possible size of the "correct" map looking penumbra. Later, shading was calculated as a fraction of the barrage of the scene geometry projected back onto the light source. Methods in most cases neglect physical correctness, so they are difficult to apply for large extended light sources.

Often, to calculate the illumination from extended light sources, various schemes for precomputing lighting on stage surfaces are used.

Lighting maps store lighting at the vertices of a triangular grid. As in the emissivity method, the scene should have a triangular grid with a density that is consistent with the lighting gradient.

The use of non-Lambert BRDF is either impossible or very limited. The main advantage of the method is the ability to visualize the lighting map on the GPU using vertex lighting.

Photon maps [9-12] preserve the illumination as a cloud of particle collision points with the scene surfaces that were released from the sources. Photon maps are a much more flexible method than lighting maps and have become very widespread in practice. However, the lighting estimate obtained using the photon map has low-frequency noise and the final collection is used to eliminate it. This is ray tracing along the hemisphere according to the visualization equation, where the integrand estimate of illumination is already taken using photon map. The final collection takes the vast majority of the calculation time.

Precomputed radiance transmission [13] preserves the transfer function at the vertices of the scene, allowing interactive visualization of global illumination, for example, with a dynamic environment map. However, it either requires long preliminary calculations, or supports only a low-frequency solution, or both.

The radiance cache allows you to speed up visualization several times, since only in a small part of the points visible through pixels, ray tracing along the hemisphere calculated the irradiation, and in at other points, it was extrapolated. In order to calculate the radiance value at a certain point, we look in the cache for already calculated suitable values for extrapolation. If extrapolation is possible, then we perform it and move on to the next point. If not, we calculate the illumination by tracing the rays across the entire hemisphere, and write the calculated value to the cache. To estimate the extrapolation error, the average harmonic distance to the nearest objects of the scene was used, which was calculated when tracing rays along the hemisphere. The smaller this distance, the greater the error and, hence, the density of points is higher. The density of points in the irradiation cache adapts to the lighting gradient, however, it is difficult to achieve the absence of artifacts, since the lighting gradient is calculated approximately.

In [14], it is proposed to use a multi-pass adaptive algorithm, which, when adding a new point to the cache, compares the average harmonic and relative deviation of irradiation at neighboring points, and reduces the radius of influence of the new point, as well as the radii of influence of points in the cache. It is clear that after this part points that were successfully extrapolated earlier are found to be extrapolated incorrectly. Therefore, several passes are needed until all the points are calculated “correctly”. Multi-pass significantly reduces the number of image artifacts, but leads to many excessive calculations, which reduces its efficiency.

In [15], the method of cache irradiation was improved by adding radiation gradients. The involvement of the gradients allows carrying out linear extrapolation or cubic interpolation of the illumination.

In [14], it was proposed to extend the method [15] to use any BRDF, not just Lambert ones. This required storing the lighting as a function of the direction on the hemisphere. For this purpose, spherical harmonics are used in [14]. In addition to data compression, this basis allowed us to quickly implement a convolution.

Spherical harmonics were chosen because the projection to the basis is fast, and the convolution is a scalar product of the coefficient vectors. This made it possible to apply not only Lambert's BRDF, but also moderately directional-diffuse BRDF; however, the method is not applicable to sharply directed BRDF.

In [16], the illumination was divided into near and far. For long-range illumination, the method of radiation cache [14] with a basis of spherical harmonics was used. Since long-range lighting changes quite smoothly, the radiation cache is much rarer than for full lighting. For near lighting, the emissivity method was used without taking into account the visibility function to calculate form factors, so near lighting is calculated very quickly –it is enough to find all the triangles near the point. Although ignoring visibility gives a physically incorrect solution, visually in most cases it seems to be “correct”. For scenes of medium complexity with diffuse surfaces, the algorithm is well suited, creating images of visual quality comparable to [15], but much faster. Work [14] describes the use of projection together with the radiance cache.

The intersection points of the rays and the scene surfaces that were previously calculated and written to the cache are projected onto the hemisphere at a new point on the surface where you need to calculate the lighting.

Reprojection is not a fast procedure, so it is necessary to normalize the vector and calculate inverse trigonometric functions (if spherical coordinates are used to parameterize the hemisphere). In addition, you need to use the z-buffer and do something with the voids that will remain after the projection. In this paper, we will try to eliminate the shortcomings of the reprojection algorithm as applied to the radiance cache.

3. PROBLEM FORMULATION

The calculation of illumination on the surface can be written down and solved by the Monte Carlo method quite simply, but the efficiency of the direct solution is low. Unbiased rendering methods, for example, path tracing are classic forms of Monte Carlo light transport [17].

For example, consider the calculation of lighting from a triangular light source. You can calculate the illumination from such a source by both the first and second equations. For a triangle, if it occupies a small part of the hemisphere, it is much easier to generate several points on its area evenly than to deal with a spherical triangle on the hemisphere. Nevertheless, if the triangle is large and one vertex is close to the point where we want to calculate the illumination, and the other two are far away, then the points generated in this way will lie strongly unevenly on the hemisphere over which we integrate. This will negatively affect convergence or, in other words, cause a large variance. Therefore, generating points directly on a spherical triangle will increase the speed and accuracy of calculations. However, if the triangle occupies a small part of the hemisphere and lies far away, then such a more complex algorithm does not give anything. This example shows that large light sources require special calculation methods. For different scenes, choose one or another equation, depending on where it is more convenient to generate points on the hemisphere or on the surfaces of light sources. The first formulation with a hemisphere integral is more convenient to apply if the incident radiation comes from most of the hemisphere. Examples are areal sources, environment maps defined with a large dynamic range, and areas of secondary lighting integration where all surfaces become light sources. This class of sources is more time-consuming to calculate because it requires a much larger sample. Often the lighting, even from the big light sources change slightly near the surface point. Then the rays coming from the light source are called coherent.

The coherence property allows you to speed up calculations. All algorithms try to use this property in one way or another.

As discussed above, there are many algorithms for calculating illumination from extended light sources, but they all have some drawbacks and are designed for a very limited number of cases. Therefore, there is a need for a more universal algorithm for calculating scenes of great complexity that have extended light sources, as well as secondary sources. Despite their complexity, such scenes often have a high level of coherence. But the difficulty lies in the fact that the same point the surface can be completely in shadow or completely in light from some light sources (rays to such sources are coherent) and is in partial shade from other sources (where the coherence of the rays is small). Therefore, simple methods of interpolation or extrapolation of lighting is not suitable. Additional difficulties arise with secondary light sources that are implicitly represented in the scene and their location is not known in advance. As the basis of the new algorithm, the irradiation (radiation) cache was chosen as the most general and scalable algorithm. Unfortunately, the irradiation (radiation) cache has drawbacks, as well as other algorithms that have been considered.

First, it is difficult to provide a given level of error. In practice, this is overcome by a much larger number of points in the cache than it could be, and by manually selecting parameters for each new scene. The most important thing in the previous radiation caches is the need for multi-pass and interpolation (extrapolations) over multiple points from the cache in areas where lighting changes rapidly or BRDF is directional.

Secondly, in order to be able to use high-frequency lighting and BRDF, as well as correct processing of the visibility function, spherical harmonics cannot be used for the basis on the hemisphere due to their low frequency, as was done in [14].

Third, the use of gradients for high-frequency radiation cache is impractical. On the one hand, because of their instability, which was mentioned earlier, and on the other because of the fundamental inability to provide nonlinear extrapolation of radiation, which is clearly manifested in high frequency BRDF and lighting. For a more accurate extrapolation, a projection was chosen that it is relatively slow, but it allows more correct processing of “spherical distortions” and non-linear extrapolation. For non-linear extrapolation, it is necessary to be able to determine in the space of the hemisphere where the background that remained unchanged, and the background that was closed earlier, as well as the object that moved, and the unknown that opened. This separation is a generalization of the idea of separation into near and far lighting, which was mentioned above.

This work aims to develop an effective visualization method based on radiance caching and reprojection.

4. GEOMETRIC OBJECTS AND OPERATIONS

The most common model for visualizing three – dimensional images is the polygonal approximation. Along with many advantages, this model has its drawbacks. By modeling real objects, an approximate polygonal model is constructed. To increase the image quality, you most often need to increase the number of polygons. An increase in the number of polygons leads to an increase in the rendering time and the amount of memory used. In addition, changing the scale of the object introduces additional problems, because you cannot quickly and efficiently change the number of polygons for the object model. You can get rid of such shortcomings by applying an analytical task of volumes and rasterizing them using ray tracing algorithms. The analytical task of volumes does not require a large amount of memory. The problem of synthesis of realistic images is relevant for various simulators, virtual studios and three-dimensional games. Now, there are already works on visualization of functionally defined surfaces, but their application is limited to a rather narrow class of surfaces and slow visualization. The algorithms used are difficult to optimize, which also imposes restrictions on practical application.

4.1. Geometric Objects

It is proposed to use a special class of objects called “free forms” [18]. Each free form represents a base surface and a perturbation on that surface. The base surface and perturbation are given by second-degree polynomials-quadrics:

$$F(x,y,z) = A_{11}x^2 + A_{22}y^2 + A_{33}z^2 + A_{12}xy + \\ + A_{13}xz + A_{23}yz + A_{14}x + A_{24}y + A_{34}z + A_{44} \ge 0, \quad (1)$$

where x, y and z are spatial variables.

The free form is

$$F'(x,y,z) = F(x,y,z) + \sum_{i=1}^{N} f_i R_i(x,y,z), \quad (2)$$

here fi is the form-factor; the perturbation function R(x, y, z) is

$$R_i(x,y,z) = \begin{cases} Q_i^3(x,y,z), if\ Q_i(x,y,z) \ge 0 \\ 0, if\ Q_i(x,y,z) < 0 \end{cases}, \quad (3)$$

where Q(x, y, z) is the perturbing quadric.

Fig. 1 shows textured opaque free form (a) and semi-transparent free form (b). To achieve smoothness, the perturbation function is raised to the third power (3).

4.2. Geometric Operations

Two main types of the set of geometric objects are simple geometric objects and complex geometric objects. A complex geometric object is a union operation on simple geometric objects [19]. If the objects G1 and G2 are defined as $f1(X) \ge 0$ and $f2(X) \ge 0$. The union of the objects G1 and G2 is the operation G3=Φ¡(G1,G2) with the definition

$$f_3 = \psi(f_1(x,y,z), f_2(x,y,z)) \ge 0, \quad (4)$$

where ψ is the function of two variables.

5. METHOD DESCRIPTION

5.1. Ray tracing

An algorithm based on the radiation cache and projection is proposed.

First, we trace the rays through the pixels; find the nearest intersection points with the scene and save them to an array. In addition to the point itself, surface normals and pointers to the BRDF are also calculated and stored. Using ray tracing, we calculate the primary illumination from point and parallel light sources. At the end, we pass through the array of pixels and request the radiation cache, if

a *b*

Fig. 1 **The free forms: (a) textured opaque free form, (b) semi-transparent free form**

***Source*: Compiled by the author**

there is a suitable point in the cache for extrapolation, then we extrapolate (we perform if not, then trace the rays along the hemisphere and create an entry in the cache (save the texture of the incident radiation and information for the projection).

In order to store the radiation falling from the hemisphere, in linear memory you need to have maps from the hemisphere to the plane and back. The mapping [20] was chosen because it is continuous, preserves area, and has a slight distortion. If the mapping from a square to a hemisphere is with density $\cos\theta/\pi$, then the integral over the hemisphere turns into an integral over the square with coordinates $u, v \in [0,1]\times[0,1]$.

5.2. Radiance caching

The input is the point on the surface where the illumination was calculated, and the normal to the surface. At the output, we have the texture of the incident radiation and an array of s-elements. The S-element is an abbreviation for surface element (analogous to the term surfel, an abbreviation for surface element). This term refers to the fact that at a point on the surface there is some flat differential platform parallel to the surface at this point.

First, we trace the rays $n \times n$ along the hemisphere (a typical value is 32x32, 64x 64), we find the point of collision with the surfaces of the scene (or infinitely removed background). Calculate the incident radiation that goes along the rays, save it to the texture. The calculated points are divided into layers by depth. Each point becomes an s-element. Sorting is performed by K layers, it is not necessary to sort s-elements inside the layer, so sorting can be performed in time $K \cdot O(n \times n)$, which is quite fast compared to ray tracing $n \times n$.

Layers are selected on a logarithmic scale, because the nearest s-elements need to be sorted more accurately. Inside the layer, we combine adjacent s-elements, having a close normal. To do this, we perform several passes, alternating passes along the rows with passes along the columns of the texture. This helps to make the s-elements less elongated. Such neighboring s-elements will be reprojected into pairs of neighboring ones. Therefore, by combining them, you can save on the future projection procedure. For the obtained n-elements, we find the corner points in u, v space. At the end, we save all the s-elements and the texture of the incident radiation to the cache.

5.3. Reprojection

The input of the algorithm is the coordinates of a point on the surface, its normal, BRDF and the viewing direction. As well as the nearest entry from the cache: the texture of the incident radiation, an array of s-elements. At the utput, we have outgoing radiation from a point on the surface along the direction of view.

The projection is carried out layer-by-layer, starting from the nearest layer and ending with the farthest. This is similar to a reverse painter's algorithm. At the same time, a full-fledged Z-buffer is not needed; you can only do with a fullness mask.

We project all s-elements at a new point. For each of them we have in u, v there are four corner points in space. In General, they form some kind of quadrilateral (perhaps not even convex). At the same time, we build a rectangle based on four points, the coordinates of the corner points of which are — these are the maximum u, v coordinates of four reprojected points. By doing this, we calculate the approximate projection of the s-element. P-element masks and unknown masks are created for the s-element. S-element mask — this is a bitmask, where one are located in the places u, v of the space where the s-element was projected. Next comes the creation of bit mask of the motion trace of the s-element projection. This mask is calculated as the enclosing rectangle u,v of the coordinates of the s-element at the point from the cache and the new projection of the s-element. The trace mask allows you to determine the area u,v of space where an unknown zone has opened. The unknown mask is obtained as a trace mask minus the fill mask. The unknown mask prohibits filling in more distant layers in the unknown area. An object that may be located in such a zone may overlap the background visible from the new point. With the mask of the unknown, we prevent the fact that the overlapped background will be taken into account incorrectly. It should be noted that the unknown mask is filled for all s-elements of the next layer after all s-elements of this layer have been processed. This prevents neighboring s-elements of the same layer from giving a reassessment of the unknown on each other.

Next, we bypass the fill mask with some step k and trace the rays where there are gaps. There are several pixels of the mask per ray. Add the radiation obtained by ray tracing, with a weight equal to the number of unfilled pixels of the mask divided by the number of pixels k^2.

The BRDF is a sum of two terms (specular and diffuse):

$$f_r(x,\vec{\omega}',\vec{\omega}) = f_{r,s}(x,\vec{\omega}',\vec{\omega}) + f_{r,d}(x,\vec{\omega}',\vec{\omega}) \; . \quad (5)$$

The incoming radiance uses three terms (direct illumination, caustics, indirect illumination):

$$L_i(x,\vec{\omega}') = L_{i,l}(x,\vec{\omega}') + L_{i,c}(x,\vec{\omega}') + L_{i,d}(x,\vec{\omega}') \; . \quad (6)$$

The reflected radiance is a sum of four integrals (we need to compute the reflected radiance from a surface):

$$\begin{aligned} &L_r(x,\vec{\omega}') = \int_{\Omega_x} f_r(x,\vec{\omega}',\vec{\omega}) L_{i,l}(x,\vec{\omega}')\cos\theta_i d\omega_i' + \\ &\int_{\Omega_x} f_{r,s}(x,\vec{\omega}',\vec{\omega})(L_{i,c}(x,\vec{\omega}') + \\ &+L_{i,d}(x,\vec{\omega}'))\cos\theta_i d\omega_i' + \\ &+\int_{\Omega_x} f_{r,d}(x,\vec{\omega}',\vec{\omega}) L_{i,c}(x,\vec{\omega}')\cos\theta_i d\omega_i' + \\ &+\int_{\Omega_x} f_{r,d}(x,\vec{\omega}',\vec{\omega}) L_{i,d}(x,\vec{\omega}')\cos\theta_i d\omega_i' \end{aligned} \; . \quad (7)$$

Caustics are evaluated by integral [8] (Fig. 2):

$$\int_{\Omega_x} f_{r,d}(x,\vec{\omega}',\vec{\omega}) L_{i,c}(x,\vec{\omega}')\cos\theta_i d\omega_i' \; . \quad (8)$$

6. IMPLEMENTATION AND PERFORMANCE

For image display is used DirectX. Testing of productivity of the offered method of realization has been made. CUDA from NVIDIA was used. Together with a set of software, she allows to realize programs in language C for execution on a GPU. Testing was performed on the GPU GTX 580.

The method gives an order of magnitude acceleration in comparison with the classical implementation of the Monte Carlo method. Figure 2 shows the result of modeling the method.

The main results are as follows.

1) Reduced scene databases by using functionally defined objects.

2) In contrast to the known radiance caching algorithms, the proposed method can works with high-frequency data.

3) In comparison with the classical implementation of the Monte Carlo method, the method gives an acceleration of ten times with comparable calculation accuracy.

4) The method is effective in calculating the final collection in photon maps and emissivity, illumination from the environment map, a given area with a large dynamic range, shadows from large area light sources, and "blurred" reflections.

5) The method uses the coherence property, which allows you to speed up the calculations.

6) The method can use non-Lambert bidirectional reflection functions.

7) The lighting estimate obtained using the proposed method does not have low-frequency noise.

8) The radiance cashing allowed several times faster visualization, since only a small part of the points visible through pixels. The radiance is calculated by tracing the rays along the hemisphere, and at other points, it is extrapolated.

Fig. 2. **A water glass with a caustic**

***Source*: Compiled by the author**

9) The method lacks multi-pass, which significantly reduces the number of image artifacts, but leads to a lot of excessive calculations, which reduces its efficiency.

10) The method is universal for calculating scenes of great complexity that have extended light sources, as well as secondary sources.

7. CONCLUSIONS

The method consists of the main parts: the top-level algorithm, the cache entry, and the projection. The method was based on the radiance cache, as the most general and scalable algorithm. Unfortunately, the radiance cache also has drawbacks. First, it is difficult to provide a given level of error. Secondly, in order to be able to use high-frequency illumination and bidirectional reflection function, as well as correct processing of the visibility function, for the basis on spherical harmonics cannot be used in the hemisphere due to their low frequency. Third, the use of gradients for high-frequency radiation cache is impractical. For a more accurate extrapolation, a projection was chosen that it is relatively slow, but it allows you to more correctly handle "spherical distortions" and perform nonlinear extrapolation. For non-linear extrapolation, it is necessary to be able to determine in the space of the hemisphere where there is a background that has remained unchanged, and a background that was closed earlier, as well as an object that has moved, and an unknown that has opened. This separation is a generalization of the idea of separation into near and far illumination.

Bitmask operations are performed using SSE2 instructions that are programmed by intrinsically. Thus, bitwise logical operations are applied to 128 bits simultaneously. The algorithm can be used to calculate the final collection in the methods of photon maps and emissivity, illumination from the environment map specified with a large dynamic range, shadows from large area light sources, "blurred" reflections. The algorithm is physically correct and can be used to render very complex scenes with high frequency BRDF and lighting. Similar caching algorithms cannot provide.

REFERENCES

1. Nikoghossian, A. G. "Invariance Principle and Bilinear Relations of the Radiative Transfer Theory. I". *The Astrophysical Journal. The American Astronomical Society*. Printed in U.S.A. January 2009; 483(2): 849–856. DOI: 10.1086/304258.
2. Nikoghossian, A. G. "Groups and Their Representations in the Theory of Radiative Transfer. II". *Astrophysics.* June 2014; 57(2): 272–286. DOI: 10.1007/s10511-014-9333-x.
3. Nikoghossian, A. G. "Groups and their Representations in the Theory of Radiative Transfer. III". *Astrophysics*. February 2019. 62(1). DOI: 10.1007/s10511-019-09567-6.
4. Peraiah, A. "An Introduction to Radiative Transfer". Methods and Applications in Astrophysics, Chapter 5 – Principle of invariance. *Cambridge University Press.* June 2012. p.112–145. DOI: https://doi.org/10.1017/CBO9781139164474.006.
5. Aila, T. & Samulii, L. "Understanding the Efficiency of Ray Traversal on GPUs". *HPG '09: Proceedings of the Conference on High Performance Graphics 2009*. 2009. p.145–149. DOI: 10.1145/1572769.1572792.
6. Ludvigsen, H. & Elster, A. C. "Real-Time Ray Tracing Using Nvidia OptiX". *Published in Eurographics. Computer Science*. 2010. p.1–4.
7. Guennebaud, G., Barthe, L. & Paulin, M. "High-Quality Adaptive Soft Shadow Mapping". *Computer graphics forum*. 2007; Vol. 26. No. 3: 525–533.
8. Hasenfratz, J.-M., Lapierre, M., Holzschuch, N. & Sillion, F. "A Survey of Real-Time Soft Shadows Algorithms". *Eurographics'03 State-of-The-Art Reports*. 2003. p.1–20.
9. Bitterli, B. & Jarosz, W. "Beyond points and beams: Higher dimensional photon samples for volumetric light transport". *ACM Transactionson Graphics, Proceedings of SIGGRAPH'17*. July 2017; 36, 4: 1–12. DOI: org/10/gfznbr.
10. Hachisuka, T., Ogaki, S. & Jensen, H. W. "Progressive photon mapping". ACM Transactions on Graphics, Proceedings of SIGGRAPH Asia. 2008; 27, 5 130:1–130:8. DOI: org/10/d8xxn3.
11. Zinke, A. & Weber, A. G. "Efficient Ray Based Global Illumination Using Photon Maps". *Conference: Vision, Modeling and Visualization (VMV 2006).* November 2006. – Available from: https://www.researchgate.net/publication/235417074_Efficient_Ray_Based_Global_Illumination_Using_Photon_Maps. – [Accessed: Dec, 2020].
12. Jarosz, W. & Nowrouzezahral, D. "A Comprehensive Theory of Volumetric Radiance Estimation Using Photon Points and Beams". ACM Transactions on Graphics. January 2011; 30(1): 5. DOI: 10.1145/1899404.1899409.
13. Kautz, J., Sloan, P. P. & Lehtinen, J. "Precomputed Radiance Transfer: Theory and Practice". *SIGGRAPH '05*. July 2005. DOI: org/10.1145/1198555.1198682.
14. Krivanek, J., Gautron, P., Pattanaik, S. & Bouatouch, K. "Radiance Caching for Efficient Global Illumination Computation". *IEEE Transactions on Visualization and Computer Graphics*. 2005; 11(5): 550–561. DOI: 10.1109/TVCG.2005.83.

15. Ward, G. J. & Heckbert, P. S. “Irradiance gradients”. *SIGGRAPH '08.* August 2008. Article No. 72. p.1–17 DOI: org/10.1145/1401132.1401225.

16. Arikan, O., Forsyth, D. A. & O'Brien, J. F. “Fast and detailed approximate global illumination by irradiance decomposition”. *ACM Transactions on Graphics*. July 2005. 24(3). DOI: org/10.1145/1073204.1073319.

17. Celestino, S., Romano, D., Laccetti, J. & Lapegna, M. “Bidirectional Path Tracing. A rendering method with Global Illumination on GPU”. *Applied Mathematical Sciences.* January 2014; Vol. 8 No. 133: 6783–6790. DOI: 10.12988/ams.2014.49694.

18. Romanyuk, O. N., Vyatkin, S. I., Mykhaylov, P. I. & Chekhmestruk, R. Y. “Interactive shape modeling using functionally defined objects”. *Herald of Advanced Information Technology*. *Publ. Science i Technical.* Odesa. Ukraine. 2020; Vol.3 No. 3: 149–163. DOI: 10.15276/hait.03.2020.4. – Available from: https://hait.opu.ua/?fetch=articles&with=info&id=63. – [Accessed: Dec, 2020].

19. Romanyuk, O. N, Vyatkin, S. I., Pavlov, S.V., Romanyuk, O. V., Snigur, A.V., Komada, P., Smailova, S. & Yeraliyeva B. “A function-based approach to real-time visualization using graphics processing units”. *Proc. SPIE: Photonics Applications in Astronomy, Communications, and Industry*. *High Energy Physics Experiments.* 2020. Vol. 11581. 115810E. Available from: https://www.scopus.com/authid/detail.uri? origin=resultslist &authorId=6603031329&zone=. DOI: 10.1117/12.2580212. – [Accessed: Dec, 2020].

20. Robertson, D.G. & J. Lee, J. “On the use of constraints in least squares estimation and control”. *Published 2002, Mathematics, Computer Science. Autom*. DOI: 10.1016/S0005-1098(02)00029-8Corpus ID: 45431348.

Conflicts of Interest: the authors declare no conflict of interest

Received 10.12.2020
Received after revision 04.03.2021
Accepted 15.03.2021

DOI: 10.15276/hait.01.2021.4
УДК 004.925.8

МЕТОД РОЗРАХУНКУ ФУНКЦІЇ ВІДОБРАЖЕННЯ ГЛОБАЛЬНОГО ОСВІТЛЕННЯ З ВИКОРИСТАННЯМ ФУНКЦІЙ ЗБУРЕНЬ

Роман Юрійович Чехместрук[1)]
ORCID: 0000-0002-5362-8796; Rc.ua@3dgeneration.com
Павло Ігорович Михайлов[2)]
ORCID: 0000-0001-5861-5970; pm@3dgeneration.com
Сергій Іванович Вяткін[3)]
ORCID: 0000-0002-1591-3588; sivser@mail.ru
[1)] 3D GENERATION UA, 21021, вул. Пирогова, 37. Вінниця. 21021, Україна
[2)] CEO 3D GNERATION GmbH, 90-92, Märkische Straße. Дортмунд, 44137, Німеччина
[3)] Інститут автоматики й електрометрії, СВ РАН. Ак. Коптюга, 1. Новосибірськ, 630090, Росія

АНОТАЦІЯ

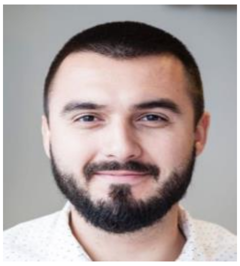
Поява нового апаратного забезпечення та постійно зростаючі вимоги до складності сцен стимулюють розробляти нові підходи до розрахунку освітлення. Сучасна візуалізація вимагає не тільки фотореалістичного, але й фізично коректного розрахунку освітлення. Ядром будь-якого алгоритму розрахунку глобального освітлення є обчислення інтегралу освітлення по півсфері. Метою роботи є розробка ефективного методу візуалізації, заснованого на кешуванні та репроекції випромінювання. У цій статті представлений модифікований метод, який усуває недоліки алгоритму репроекції для кешу випромінювання. Репроекція є трудомісткою процедурою, оскільки необхідно нормалізувати вектори та обчислити обернені тригонометричні функції, якщо для параметризації півсфери використовуються сферичні координати. Крім того, необхідно використовувати z-буфер і вирішувати задачу з порожнинами, які залишаться після проекції. До того ж, для розрахунку освітлення з віддалених джерел відомі алгоритми мають певні недоліки та розраховані на дуже обмежену кількість випадків. Тому в цій роботі розроблено універсальний алгоритм для обчислення сцен великої складності, що мають віддалені джерела світла, а також вторинні джерела. Складність полягає в тому, що одна і та ж точка поверхні може бути розмішена як повністю в тіні або повністю у світлі від деяких джерел світла (промені до таких джерел є когерентними), так і в півтіні від інших джерел (де когерентність променів невелика). Тому прості методи інтерполяції або екстраполяції освітлення непридатні. Додаткові труднощі виникають із вторинними джерелами світла, які неявно представлені на сцені, і їх місце розташування не відомо заздалегідь. Запропонований метод кешує функцію падаючого випромінювання і використовує обчислені значення в сусідніх точках поверхні, що значно зменшує кількість шляхів променів і обчислення функції відображення. На відміну від інших алгоритмів кешування випромінювання, запропонований спосіб може працювати з високочастотними даними. У порівнянні з класичною реалізацією методу Монте-Карло, метод дає прискорення на порядок із співставною точністю розрахунку. Метод може бути використаний для обчислення кінцевого збору в методах фотонних карт та випромінювальної здатності, освітлення з набору карти оточення з великим динамічним діапазоном, тіней від джерел світла великої площі, «розмитих» відображень тощо.

Ключові слова: функції збурень; геометричні об’єкти; глобальне освітлення; випромінювання; відображення; фотонні карти; трасування променів

ABOUT THE AUTHORS

Roman Yurievich Chekhmestruk – Candidate of Technical Sciences, Technical Director
3D GENERATION UA. 37, Pirogova Str. Vinnitsa, 21021, Ukraine
ORCID: 0000-0002-5362-8796; Rc.ua@3dgeneration.com
Research field: Formation and Processing of Graphic Images

Роман Юрійович Чехмейструк – кандидат технічних наук, технічний директор
3D GENERATION UA, вул. Пирогова, 37. Вінниця, 21021, Україна
Область наукових інтересів: формування та обробка графічних зображень



Pavlo Igorovich Mykhaylov – general director CEO 3D GNERATION GmbH (Germany)
90-92, Märkische Straße. Dortmund, 44141, Germany
ORCID: 0000-0001-5861-5970; pm@3dgeneration.com
Research field: Formation and Processing of Graphic Images

Павло Ігорович Михайлов – генеральний директор CEO 3D GNERATION GmbH (Німеччина).90-92,
Märkische Straße. Дортмунд, 44137, Німеччина
Область наукових інтересів: формування та обробка графічних зображень

Sergey Ivanovich Vyatkin – Candidate of Technical Sciences, senior scientific
Researcher Institute of Automation and Electrometry, SB RAS. 1, Ak. Koptyuga,
Novosibirsk, 630090, Russia ORCID: 0000-0002-1591-3588; sivser@mail.ru
Research field: Formation and Processing of Graphic Images

Сергій Іванович Вяткин – кандидат технічних наук, старший науковий співробітник Інституту
автоматики й електрометрії, СВ РАН, Ак. Коптюга, 1, Новосибірськ, 630090, Росія
Область наукових інтересів: формування та обробка графічних зображень