# Approximate global illumination using photon mapping: A review

Haochen Ding

Beijing University of Technology Beijing, China

dddhc01@163.com

**Abstract.** This paper introduces the two-step process of the standard photon mapping algorithm and a variety of other global biased illumination algorithms using photon mapping that have been proposed in recent years. Since there are many global biased illumination algorithms based on photon mapping, this review is a combination of these algorithms for the convenience of the reader, and the two-step process of standard PM will also be presented in this review to accommodate readers who are not familiar with the field. We will first introduce the specific implementation process and principles based on the two steps of PM, photon tracking and rendering, and then introduce a variety of polarized illumination techniques using photon maps, compare these methods and discuss the advantages, disadvantages, and applicability of each method in detail, and finally give a specific implementation of each method to give readers a clearer understanding of these methods. The final implementation of each method is shown to give the reader a clearer understanding of these methods.

**Keywords:** photon, illumination, mapping.

## 1. Introduction

Rendering is an extremely important field in 3D computer graphics, and the simulation of global illumination under the general environment has been an important research topic, which refers to a rendering technique that considers both the direct light from the light source in the scene and the indirect light after the reflection of other objects in the scene. After decades of development, global illumination nowadays has a variety of implementation directions, the main schools of global illumination are Ray tracing, Path tracing, Photon mapping, Point Based Global Illumination, Radiosity, Metropolis light transport, Metropolis light transport, Spherical harmonic lighting, Ambient occlusion, Voxel-based Global Illumination, Voxel-based Global Illumination Voxel-based Global Illumination, Light Propagation Volumes Global Illumination, Deferred Radiance Transfer Global Illumination, Deep G-Buffer based Global Illumination, etc. And Photon mapping is an important branch of global illumination.

Photon mapping, like ray tracing, is a rendering method in the field of realistic rendering. It can be a good solution to the caustic phenomenon which is relatively difficult to handle by traditional ray tracing type methods. The Photon mapping algorithm was first proposed and implemented by Jensen in '96 and is the basis of the entire algorithm logic, which has evolved and modified in the intervening years, resulting in a series of new and improved implementations, such as Final Gathering, irradiance caching, photon ray splatting, Progressive Photon Mapping (PPM), Stochastic Progressive Photon Mapping (SPPM) algorithms. These methods have been widely used in various fields, especially in film rendering, game engines, game production, etc. Some of them have become industry-standard algorithms.

Since the development of photon mapping technology in the field of computer rendering, testers are applying it to a wide variety of complex scene surface simulations to pursue realistic light effects. The original photon mapping algorithm was designed to solve the problem of traditional ray tracing algorithms in which the surface properties are relatively homogeneous, making it difficult to enrich the multiple optical effects that occur after light touches the surface of an object and diffuse reflections can't be simulated correctly. However, with the gradual application of photon mapping, the problems of this algorithm itself have come to the fore, for example, in some places where the photon density is not sufficient to produce a large amount of low-frequency noise, the spherical estimation will be too much, and the disc estimation will be too dark. Therefore, the resulting Final Gather improvement method is to solve the problem of low-frequency noise generated by insufficient photon density to obtain better rendering quality, but still can not eliminate the impact of boundary bias and topological bias, while taking a huge amount of time. The corresponding irradiance caching is a further optimization of FG because the illuminance changes slowly in the neighborhood, we can use the illuminance caching to simplify the calculation so that we do not need to use the costly FG for each point. Photon ray splatting can better solve the boundary bias and topological bias, but it still has bias and requires a large number of photon emissions to solve. SPPM is an improvement of PPM, with the advantage that the effect of simulating smooth reflections is stronger than PPM, and the disadvantage that the algorithm is not as efficient as PPM.

In short, each of these algorithms has its advantages and disadvantages, and the applicable scenarios and fields are different. To provide users with a better understanding of the principle implementation, advantages and disadvantages, and specific application scenarios of these algorithms, we have collected articles in the field of algorithms related to photon mapping in recent years. In addition, we have classified the different photon mapping algorithms.

Major contributions:

Collection of related algorithms and evaluations derived from photon mapping over the years.

The collected algorithms are divided into 6 categories, which are traditional photon mapping models, Final Gathering, irradiance caching, Photon ray splatting, Progressive Photon Mapping (PPM), and Stochastic Progressive Photon Mapping (SPPM).

The principle of each algorithm, the implementation process, the corresponding advantages and disadvantages, and the applicable scenarios, and listed in the form of a table.

The results of these algorithms are compared for better visual understanding and are presented in tabular forms.

A summary evaluation of the effectiveness of these algorithms.

This survey is organized as follows. First, Section 2 introduces the traditional photon mapping model, Section PHOTON MAPPING-RELATED ALGORITHMS presents the principles of the process of implementing the remaining five algorithms as well as their advantages and disadvantages. Section TABLES discusses and compares the different effects produced by these algorithms and evaluates them, and finally, Section CONCLUSION summarizes these algorithms.

## 2. Photon mapping-related algorithms

### 2.1. Classical photon mapping model

The classical photon mapping model is an algorithm implemented by Jensen in '96 for computing global illumination. It consists of two main steps, photon tracing, and ray tracing.

The first step, photon tracing, can be divided into three small steps, which are photon emission, photon diffusion, and photon storage, in that order.

#### 2.1.1. Photon Tracing

1) Photon Emission

During the photon emission phase, the light source produces a number of photons, each representing a fraction of the total power. Each photon contains a position and direction xp, wp, and an associated intensity. Because each light in the scene can be emitted in a limited number, when the scene is sparse objects will lead to many photons that can not hit any object. To make efficient use of photons, you can use the Projection Map to filter the direction of photon emission.

The Projection Map approach is to observe the scene from the perspective of the light source, the light source is broken down into many small equivalent cells, and when there is an object in the scene, the Cell is marked as "On" state, otherwise, the Cell is "Off "state, in the direction of the photon emission, only a limited number of photons to mark the "On" cell can be emitted. The main parameter of each emitted photon is flux instead of Radiance, which is the luminous flux. This value is usually used to characterize the emitted power, and the value can be obtained by dividing the Flux of the light source by the number of emitted photons.

The transmitting power is

$$\frac{\Phi}{N} \tag{1}$$

Φ stands for luminous flux

N stands for the number of emitted photons.

2) Photon Diffusion

Next is the photon diffusion stage, when the photon is generated from the light source and emitted, it will collide and intersect with the surface in the scene and then be diffusely reflected, reflected by the mirror, refracted, or absorbed.

The simulation of a realistic situation requires the use of a large number of photons. To optimize the photons and simplify the number of photons to be used, the Russian Roulette algorithm is used for optimization. The Russian Roulette algorithm does not split the photon into reflective and refractive parts, etc., and the energy carried by the photon does not decay, instead using the percentage of photon reflection, decay as the probability of reflection and refraction - the photon is either diffusely reflected in its entirety, mirrored in its entirety, or absorbed.

3) Photon Storage

In this step, each photon that hits the non-specular surface is stored in a data structure called photon map, which is a balanced KD-Tree, ensuring that the corresponding photon is found very quickly, making the search more efficient and reducing the memory requirements of each photon. For specular reflections, the photons are not stored in the data structure but can be computed in the next rendering steps by Monte Carlo ray tracing.

The format of the pixel data in the photon map is shown in the following figure:

```
struct photon
{
   float x,y,z;//collision position
         char p[4];//RGBE encoded light flux
   char phi, theta;//direction of incident light
   short flag;//data for the KD-Tree structure
}
```

### 2.1.2. Ray Tracing

1) Direct approximation estimation using photon map

In the rendering process, we usually use an approximation of the rendering equation to calculate the reflected rays leaving the surface of the object. The rendering equation is shown below:

$$L_r(x,\vec{\omega}) = \int_{\Omega} f_r\left(x,\overrightarrow{\omega'},\vec{\omega}\right) L_i\left(x,\overrightarrow{\omega'}\right) \cos\theta_{\beta}\, d\overrightarrow{\omega'} \tag{2}$$

Since the formula in (2) calculates the radiance, we can relate it to the flux stored in the photon map, due to the:

$$L(x,\vec{\omega}) = \frac{d\Phi^2(x,\vec{\omega})}{d\omega\cos\theta dA} \tag{3}$$

By combining the two equations (2) (3) we can finally arrive at

$$L_r(x,\vec{\omega}) = \sum_{p=1}^{n} f_r\left(x,\overrightarrow{\omega_p},\vec{\omega}\right) \frac{\Delta\Phi_p\left(x,\overrightarrow{\omega_p}\right)}{\pi r^2} \tag{4}$$

n is the number of nearest photons (the distance of the nearest photons is usually counted to calculate the incident light), and

$\phi_p$ denotes the power of the pth photon, and $f_r$ denotes the BRDF

$\omega, \omega_p$ denotes the outgoing and incoming directions, and r denotes the radius of the sphere containing n vertices.

This integral can be calculated by using the photon map for approximate direct estimation, which is the KNN calculation method. With the target point as the center of the sphere, the nearest k photons are selected as samples, and the radius of the sphere is set equal to the distance of k th photons from x, photon density = k/(nV), k is the number of photons, n is the total number of photons, and V is the volume of the enclosing sphere.

2)Rendering equation improvements

The rendering equation in (2) is a very expensive method, so we can improve the rendering equation in (2) by using a combination of photon maps and BRDF to obtain the following rendering equation, which is divided into several parts of the sum.

$$L_r = \int_\Omega f_r L_{i,l} \cos\theta_i \, d\omega_i + \int_\Omega f_{r,s}\left(L_{i,c} + L_{i,d}\right) \cos\theta_i \, d\omega_i + \int_\Omega f_{r,d} L_{i,c} \cos\theta_i \, d\omega_i + \int_\Omega f_{r,d} L_{i,d} \cos\theta_i \, d\omega_i$$

where $f_r = f_{r,s} + f_{r,d}$and $L_i = L_{i,l} + L_{i,c} + L_{i,d}$

The simplified expression is

$$L_r = L_{\text{direct}} + L_{\text{specular}} + L_{\text{caustics}} + L_{\text{indirect}} \tag{5}$$

This equation(5) is also used to calculate the radiance of the light leaving the surface of the object, where the calculation of the direct light part and the specular reflection part can be rendered by the high-precision Monte Carlo ray tracing method, while the calculation of the scattered light and the indirect light part can be directly estimated by using the photon map (see section 2.1 for details).

(a) Direct light and specular reflection

Terms 1 and 2 in (5) can be calculated accurately using Monte Carlo ray tracing, and the rendering calculation can be achieved using a limited number of sample rays through BRDF-based importance sampling.

(b) Caustic light

The third term in (4) is almost impossible to simulate the focal ray using the conventional Monte Carlo ray tracing method, but we can use the photon information stored in the photon map to make a direct approximation.

(c) Indirect light

The fourth term in (4) is more difficult to handle indirect illumination correctly. However, it is possible to use a photon map as a direct approximation as well. Although the direct approximation can also give good results, it can be better estimated using Final Gathering (described in detail in Section 3), but the disadvantage is that it is time-consuming and needs to generate a very large number of sampled rays, which consumes a huge amount of arithmetic power. To solve these problems, Ward developed the irradiance caching (described in detail in Section 4) method to compute global illumination, since the illumination changes slowly and continuously in the neighborhood and can be cached and reused. This eliminates the need for costly Final Gathering sampling calculations at each point, saving time significantly.

(3) Conclusion

The rendering formula can be divided into four parts, direct lighting, specular reflection, caustic and indirect lighting, that is, (4) formula for dealing with different light situations in the scene, we can take the appropriate approach, for example, specular reflection can take Monte Carlo ray tracing for sampling, while focal dispersion and indirect lighting can make photon map direct approximation, but also the use of final gathering and irradiance for indirect approximation.

### 2.2. Final Gathering

#### 2.2.1. Introduction

Final gathering is a common indirect illumination calculation means to optimize the quality of illumination calculation in photon mapping. Final gathering adds more samples to the irradiance cache to improve the accuracy and fidelity of the lighting. It is commonly used in the rendering phase of photon mapping to calculate indirect illumination.

#### 2.2.2. Implementation process

The final gathering discards the traditional photon mapping method of directly using KNN proximity estimation near the sampling points and instead emits a large number of FG sampling rays in the direction of the upper hemisphere region at each sampling point; this type of final gathering point grid can be adapted to the scene. For example, fewer final gathering points are needed in large areas, but more are needed in detailed areas and corners; thus, fast and accurate calculations can be guaranteed. After each ray collides with the diffuse surface of the scene, the KNN algorithm is used to obtain the light brightness and color of the collision point, since several rays will intersect this point, and for this reason, the sampling results are averaged so that the rendering of indirect light can be well achieved indirectly through the photon map.

#### 2.2.3. Advantages and disadvantages

The advantage of final gathering is that the number of photons stored in the indirect light reflection is small, while the direct approximation simulation will lead to a large low-frequency error, through Final gathering can use a smaller number of photons to obtain good results, but the cost is because each sampling point to send a lot of sampled rays, the time cost consumption is huge, but also still can not eliminate the effect of boundary bias and topological bias.

### 2.3. Irradiance caching

#### 2.3.1. Introduction

Irradiance caching is also a commonly used indirect illumination calculation method for optimizing the speed and quality of illumination calculation in photon mapping. Because Final Gathering requires the use of many sampled rays and is time-consuming, irradiance caching is used to reduce the cost consumption. Because the illuminance changes continuously and slowly in the neighborhood, it is not necessary to evaluate the illuminance for each point individually but only for a small part of it.

#### 2.3.2. Implementation process

The illuminance evaluation for most of the points is calculated by interpolating the irradiance gradient using the points for which the illuminance has been evaluated. In this process, each surface point in the scene is sampled, its contribution to the surrounding illumination is calculated, and the results are stored in Caching points.

$$E_i(x) = \frac{\sum_i w_i(x) E(x_i)}{\sum_i w_i(x)}$$

Gradient interpolation calculation formula (5)

$E_i(x)$ denotes the irradiance gradient information of a point x in the scene.

$E(x_i)$ denotes the irradiance gradient information of a point i in the scene.

$w_i(x)$ denotes the weight of the influence of point i on point x, which can be calculated based on distance and other factors.

The specific calculation process is as follows:

1) Build irradiance cache: at each point in the scene, its irradiance information is computed and stored in the cache. Specifically, algorithms such as photon mapping can be used to calculate the irradiance information at each point.

2) Query irradiance cache: When you need to calculate the irradiance information of a point during ray tracing, you can first query the irradiance cache to see if the irradiance information of that point has already been calculated. If it has been calculated, the irradiance information of the point is read directly from the cache and applied to the current point. If it has not been calculated, then the irradiance information for that point needs to be recalculated. To be able to correctly calculate the illumination contribution of a surface point, the normal information of the surface point needs to be considered. Specifically, a weighted average of the surrounding light based on the normal direction of the surface point is required to obtain the irradiance information of the surface point. Therefore, in the process of photon mapping (photon mapping), the normal information of the surface point needs to be recorded together.

3) Update irradiance cache: During ray tracing, if the irradiance information of the current point changes, for example, if a new light source is placed at the point or an object is moved, the irradiance information of the point needs to be updated. Specifically, the irradiance information for that point can be recalculated and updated to the cache.

### 2.3.3. Irradiance translation and rotate gradients

Irradiance translation and rotation gradient is a method used in the Irradiance Caching algorithm to calculate the irradiance of a surface point. In this method, the irradiance of a surface point is approximated by translating and rotating the Caching point.

Specifically, suppose we need to calculate the irradiance of a surface point, we first need to find the closest Caching point to it. Then, for each Caching point, the distance from it to the surface point and the angle between its normal direction and the direction normal to the surface point can be calculated. This information can be used to calculate a weighting value that is used to weigh the irradiance contribution of that Caching point to the surface point.

In the Irradiance translation and rotation gradient method, we consider that the irradiance of a Caching point varies in a local plane that is defined by that Caching point as well as its neighboring Caching points. Therefore, when calculating the weights, we use the distance and angle between the Caching points for weighing.

In addition, to more accurately approximate the irradiance of the surface points, we can also use the irradiance gradient information between the Caching points. Specifically, we can calculate the irradiance difference between Caching points and convert it into a rotational gradient for approximating the irradiance of the surface points. In addition, the irradiance gradient information between the Caching points can be used to adjust the normal direction and position of the surface points to obtain more accurate irradiance calculation results.

## 2.4. Photon Ray Splatting

### 2.4.1. Introduction

Photon Ray Splatting is also an algorithm that uses simulated global illumination, proposed and implemented by Ralf Habel and Michael Wimmer in 2009. While traditional photon mapping algorithms perform photon tracing first and then eye tracing, Photon mapping reverses the two orders, i.e. eye tracing first and then photon tracing.

### 2.4.2. Implementation process

1) eye tracing: a number of rays are emitted from the viewpoint, sampling the pixel points on the view plane

2) photon tracing: the scene light source emits multiple photons and records their paths and energies in the scene

3) Photon storage: the generated photon paths are sorted by spatial and directional domains and stored in a 5D kd tree.

4) Photon splatting kernel width calculation: Each photon ray is given a splatting kernel width, which is calculated based on the entire photon path.

5) Photon splatting: all photons are splatted one by one to positions near the line-of-sight sample. For each photon found, a function called splatting kernel is used to calculate its contribution to the color value of the point. A technique called photon differential splatting is also used, which reduces the noise and blur generated during splatting.

6) Image synthesis: the colors on all sight samples are weighted and averaged to obtain the final image.

### 2.4.3. Splatting kernel function

The splatting kernel function is a function used to calculate the contribution of photons to the color value of the intersection point, which can determine the size and shape of the contribution based on the distance, direction, and energy between the photons and the intersection point. splatting kernel function has different forms, such as Gaussian function, conical function, Epanechnikov function, etc. Different forms of the splatting kernel function will affect the quality and efficiency of the rendering results, so it is necessary to choose the right splatting kernel function according to different scenes and needs.

In general, the splatting kernel function can be expressed in the following form:

$$K(d) = \frac{1}{\pi r^2} f\left(\frac{d}{r}\right) \tag{6}$$

where K(d) is the splatting kernel function value, d is the distance between the photon and the intersection point, r is the lookup radius, and f(x) is a normalized kernel function that satisfies the following conditions:

$$f(0) = 1$$

$$\mathrm{f(x)} \geq 0$$

$$\int_0^1 f(x) d\,x = 1$$

### 2.4.4. Photon differential splatting

Photon differential splatting is a technique used to render refracted and reflected light in a way that reduces the noise and blurring generated during photon splatting. An inherent problem with photon splatting is that each photon is unaware of the other photons around it when it is splatted, which means that the size of the splat is usually determined based on heuristics rather than knowledge of the local luminous flux density. photon differential splatting uses photon differentiation to determine the size and shape, thus enabling adaptive anisotropic luminous flux density estimation.

Specifically, photon differential splatting consists of the following steps:

1) calculate the distance, direction, and energy change rate between two neighboring points on each photon path, which are the photon differential splatting.

2) Calculate the splatting kernel width at each point of the path based on the differential information of all the points on the path.

3) Sort all the photons by spatial and directional domains and store them in a 5D kd tree.

4) emit a large number of lines of sight from the camera location and sample the color and depth in the scene.

50 For each line of sight sample, find several photon paths with their nearest neighbors or closest to their directional domain and weigh the energy at their corresponding points by their splatting kernel width to obtain the final color.

### 2.4.5. Advantages and disadvantages

The photon ray splatting technique has several advantages:

- It can efficiently utilize GPU parallel computing power, thus increasing rendering speed
- It can adaptively adjust the splatting kernel width to match the local luminous flux density, thus reducing noise and blur.

- It can handle complex refraction and reflection effects, as well as shadows and indirect lighting in dynamic scenes.

The photon ray splatting technique also suffers from several disadvantages:

- It requires a large amount of memory space to store photon paths and differential information.
- It is sensitive to the complexity of the geometry and material types in the scene, which can lead to performance degradation or visual quality degradation.
- It is difficult to handle bright areas or strongly reflective areas in the scene, which may result in overexposure or jaggedness.

### 2.4.6. Applicable scenarios

- Static or dynamic scenes that require high-quality global lighting effects.
- Material objects such as glass that require complex refraction and reflection effects.
- Scenes that require a combination of high and low-frequency lighting effects.

## 2.5. Progressive photon mapping

### 2.5.1. Introduction

Progressive photon mapping was proposed and implemented by Toshiya Hachisuka, Wojciech Jarosz, and Henrik Wann Jensen in 2008. Unlike the traditional photon mapping algorithm, which also performs eye tracing before photon tracing, progressive photon mapping does not require pre-determining the size of the kernel function or storing all photon information but uses a progressive approach to gradually improve the accuracy and quality of the density estimation. quality. Specifically, each time a certain number of photons are tracked, a density estimate is performed for all line-of-sight samples, and the kernel function size and color values are updated. This process can be repeated many times, i.e., photon tracing can be repeated over and over again until satisfactory results are achieved or a preset time or memory limit is exceeded.

### 2.5.2. Implementation process

The basic implementation process is as follows:

1) eye tracing: send some rays from the viewpoint to find the scene surface (observation point) that can be observed by each pixel lens in the scene

2) for each light path, store the surface vertices of the non-specular part of the light path

3) Record the spatial coordinates x of this part of the vertices, the ejection direction \omega, the deflation term (which contains the BRDF term and the pixel filter value), and the pixel coordinates associated with the vertices. In addition, some additional data need to be stored for the Progressive Radiance Estimate, including the photon radius, the intercepted energy, and the number of photons in the radius.

4) Photon Tracing Passes: emit a certain number of photons from the light source and create a photon map to store the photons

5) Traverse each observation point and find the photons within the radius, the progressive radiosity algorithm will improve the calculation result by the new photons of this pass, and then render the result

6) Discard these data for the next pass, and repeat steps 4-5 until enough photons have been accumulated to make the image perform well enough.

### 2.5.3. Progressive Radiance Estimate

Progressive Radiance Estimate is an algorithm used in Progressive Photon Mapping (Progressive Photon Mapping) that guarantees convergence and bounded memory usage for Progressive Photon Mapping. It can handle situations that are difficult for other algorithms to handle, such as scenes that contain participating media and specular interfaces, and real light sources that are surrounded by refractive and reflective materials.

The core idea of Progressive Radiance Estimate reduces the radius of the observation point disc by increasing the cumulative number of photons, thus achieving the infinitely small radius and infinitely high density of Equation (7).\

$$L(x) = \frac{D(x)F(x)}{A} \tag{7}$$

The specific process is as follows:

-Radius Reduction:

After each photon mapping, the radius of the observation point disk is reduced according to the accumulated number of photons, so that the radius in Eq. (2) tends to zero and the density tends to infinity.

-Flux Correction:

After each photon mapping, the energy distribution on the observation point disk is corrected according to the new photons added and the old ones, so that the energy in Eq. (3) tends to be constant.

-Radiance Evaluation:

After each photon mapping, the radiance (Radiance) is calculated based on the energy distribution and surface normals on the observation point disc.

### 2.5.4. Advantages and disadvantages

Advantages of PPM over Photon Mapping:

- It can handle complex geometry and materials without precomputing or storing radiation cache. For example, it can handle dynamic scenes or participating media.
- It can reduce memory consumption by storing only one photon map instead of multiple photon maps for different bounces. For example, it can handle large scenes or high-resolution images.
- It can improve rendering quality by using an adaptive radius estimation technique that reduces noise and bias. For example, it can handle difficult lighting situations or high-frequency details.

PPM also has disadvantages compared to Photon Mapping:

- It requires more computation time due to the progressive nature of the algorithm. For example, it may take longer to converge to a satisfactory result.
- It may introduce artifacts due to the dependence of the radius estimation technique on pixel resolution and sampling density. For example, it may lead to blurring or jagged effects if the resolution or sampling density is too low or too high.

### 2.5.5. Applicable scenarios

- Suitable for scenes that require high precision and realistic global lighting effects.
- Suitable for scenes with complex geometry and materials.

## 2.6. Stochastic Progressive Photon Mapping

### 2.6.1. Introduction

Stochastic Progressive Photon Mapping is an improved global lighting algorithm proposed by Hachisuka and Jensen in 2009, which is a simple extension of the traditional Progressive Photon Mapping (PPM) algorithm.

### 2.6.2. Differences and changes

Their main differences are:

- PPM emits one ray at a time from the camera, then emits multiple rounds of photons from the light source, and updates the color and radius of the observation point (hitpoint) in the scene.
- SPPM emits multiple rays from the camera at a time and randomly selects one as the primary ray, then emits multiple rounds of photons from the light source, and updates the color and radius of the observation point (pixel) corresponding to the primary ray.

The main changes are:

- SPPM does not need to record the observation point map of the whole scene, but only the observation points corresponding to each pixel, which can save memory space and computation time.

- SPPM can converge to the correct result faster because it uses a pixel-based computation instead of a scene-based one.

### 2.6.3. Advantages and disadvantages

Some advantages of SPPM over PPM:

- It can converge to the correct result faster because it uses a pixel-based calculation instead of a scene-based one.
- It can handle cases with a high sampling rate (spp) and complex geometry (e.g. hair, grass, etc.) without noise or bias.
- It does not need to record the observation point map of the whole scene, but only the observation point corresponding to each pixel, which saves memory space and computation time.

Some disadvantages of SPPM over PPM:

- It does not handle smooth or specular reflective surfaces well because it updates only the observation points corresponding to the main rays and ignores those corresponding to the other rays.
- It needs a suitable initial radius parameter to control the number of photons around each observation point in the photon map. If the initial radius is too large or too small, it will affect the rendering effect and speed.

### 2.6.4. Applicable scenarios

- Suitable for rendering objects or materials with a high level of detail and realism (e.g. hair, grass, glass, etc.)
- Suitable for scenes with complex geometry and materials that are difficult to handle by other methods.

## 3. Conclusion

Photon mapping is a powerful global lighting algorithm that contains a series of innovative techniques to achieve high-quality and efficient image rendering. This review paper provides a detailed introduction and analysis of photon mapping-related algorithms, applications, and advantages and disadvantages.

First, we review the basic principles and algorithmic processes of photon mapping. Second, we explore some advantages and disadvantages of photon mapping, such as its ability to handle different lighting phenomena, such as global illumination, scattering, absorption, and reflection, but its algorithms are usually complex and computationally expensive.

Then, we introduce other algorithms extended by photon mapping and their advantages and disadvantages, such as progressive photon mapping, photon ray splatting, etc. The applicability of photon mapping extended algorithms is also discussed , as shown in Table I.

**Table I** advantages and disadvantages of photon mapping algorithm

| Algorithm | Comparison | | | |
|---|---|---|---|---|
| | Introduction | Advantages | Disadvantages | Applicability |
| photon mapping | Fast calculation of focal dispersion and soft indirect illumination | Simulating complex lighting effects | Generating noise and artifacts | Static scenes and high-quality rendering |
| final gathering | A method combining photon mapping and irradiance caching | Eliminating noise and artifacts in photon maps | Causing interpolation and sampling errors | High precision and low noise rendering |
| irradiance caching | A method for accelerating indirect light computation in Monte Carlo ray tracing | Significantly reducing the number of samples and computation time | Generating artifacts or distortion | Large smooth surfaces or scenes with low-frequency variations |
| Photon ray splatting | A rasterization-based method for implementing the density estimation step in progressive photon mapping | Fast splatting at any depth level | Additional camera path generation required | Surfaces of dynamic objects or objects with complex geometry |
| progressive photon mapping | A multi-step ray tracing method | Dynamically adjusting sampling density and radius | Generating noise or deviation | High-quality and real-time feedback rendering |
| stochastic progressive photon mapping | An improved method for progressive photon mapping | Improving rendering quality and stability | Requiring more memory and computing resources | Scenes with high contrast or sufficient detail |

Finally, we provide an outlook on the future of photon mapping, including its development in programmable shading, deep learning, etc. We believe that photon mapping will continue to evolve in the future and become one of the key technologies in the field of computer graphics.

## References

[1] Christensen P H, Batali D. An Irradiance Atlas for Global Illumination in Complex Production Scenes[C]//Rendering Techniques. 2004: 133-142.

[2] Ward G J, Heckbert P S. Irradiance gradients[R]. Lawrence Berkeley Lab., CA (United States); Ecole Polytechnique Federale, Lausanne (Switzerland); Technische Hogeschool Delft (Netherlands). Dept. of Technical Mathematics and Informatics, 1992.

[3] Jensen H W, Arvo J, Dutre P, et al. Monte Carlo ray tracing[C]//ACM SIGGRAPH. 2003, 5.

[4] Kontkanen J, Räsänen J, Keller A. Irradiance filtering for Monte Carlo ray tracing[C]//Monte Carlo and Quasi-Monte Carlo Methods 2004. Springer Berlin Heidelberg, 2006: 259-272.

[5] Cline D, Talbot J, Egbert P. Energy redistribution path tracing[J]. ACM Transactions on Graphics (TOG), 2005, 24(3): 1186-1195.

[6] Hachisuka T, Ogaki S, Jensen H W. Progressive photon mapping[M]//ACM SIGGRAPH Asia 2008 papers. 2008: 1-8.

[7] Xu X , Wang L , Xu Y ,et al.Distributed Global Illumination Method Based on Photon Mapping[J].IEEE Computer Society, 2016.DOI:10.1109/IIKI.2016.68.

[8] Knaus, Claude, and Matthias Zwicker. "Progressive photon mapping: A probabilistic approach." ACM Transactions on Graphics (TOG) 30.3 (2011): 1-13.

[9] Chen J, Wang B, Yong J H. Improved stochastic progressive photon mapping with metropolis sampling[C]//Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2011, 30(4): 1205-1213.

[10] Herzog R, Havran V, Kinuwaki S, et al. Global illumination using photon ray splatting[C]//Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2007, 26(3): 503-513.

[11] László S K. Monte-carlo methods in global illumination[J]. Available electronically from http://www. iit. bme. hu/szirmay/puba. html. Book Preprint, 2000.

[12] Ward G J, Rubinstein F M, Clear R D. A ray tracing solution for diffuse interreflection[C]//Proceedings of the 15th annual conference on Computer graphics and interactive techniques. 1988: 85-92.

[13] Kajiya J T. The rendering equation[C]//Proceedings of the 13th annual conference on Computer graphics and interactive techniques. 1986: 143-150.'

[14] Jarosz W, Zwicker M, Jensen H W. The beam radiance estimate for volumetric photon mapping[M]//ACM SIGGRAPH 2008 classes. 2008: 1-112.

[15] Jensen H W. Global illumination using photon maps[C]//Rendering Techniques' 96: Proceedings of the Eurographics Workshop in Porto, Portugal, June 17–19, 1996 7. Springer Vienna, 1996: 21-30.

[16] Jarosz W, Jensen H W, Donner C. Advanced global illumination using photon mapping[M]//ACM SIGGRAPH 2008 classes. 2008: 1-112.

[17] Herzog R, Havran V, Kinuwaki S, et al. Global illumination using photon ray splatting[C]//Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2007, 26(3): 503-513.

[18] Jarosz W, Nowrouzezahrai D, Thomas R, et al. Progressive photon beams[J]. ACM Transactions on Graphics, 2011, 30(6): 1-12.

[19] Marco J, Guillén I, Jarosz W, et al. Progressive transient photon beams[C]//Computer graphics forum. 2019, 38(6): 19-30.

[20] Havran V, Herzog R, Seidel H P, et al. Fast final gathering via reverse photon mapping[C]//Untitled Event. Blackwell, 2005: 323-333.

[21] Schregle R. The RADIANCE Photon Map Extension User Manual[J]. 2016.

[22] Jarosz W, Zwicker M, Jensen H W. The beam radiance estimate for volumetric photon mapping[M]//ACM SIGGRAPH 2008 classes. 2008: 1-112.

[23] Suykens F, Willems Y D. Density control for photon maps[C]//Rendering Techniques 2000: Proceedings of the Eurographics Workshop in Brno, Czech Republic, June 26–28, 2000 11. Springer Vienna, 2000: 23-34.

[24] Kaplanyan A S, Dachsbacher C. Adaptive progressive photon mapping[J]. ACM Transactions on Graphics (TOG), 2013, 32(2): 1-13.

[25] Jensen H W, Christensen P H. Efficient simulation of light transport in scenes with participating media using photon maps[C]//Proceedings of the 25th annual conference on Computer graphics and interactive techniques. 1998: 311-320.

[26] Christensen P H. Faster photon map global illumination[J]. Journal of graphics tools, 1999, 4(3): 1-10.

[27] Vorba J. Bidirectional photon mapping[J]. Proceedings of CESCG, 2011: 25-32.

[28] Havran V, Bittner J, Seidel H P. Ray maps for global illumination[M]//ACM SIGGRAPH 2004 Sketches. 2004: 77.

[29] Yu T T, Lowther J, Shene C K. Photon mapping made easy[C]//Proceedings of the 36th SIGCSE technical symposium on Computer science education. 2005: 201-205.

[30] Larsen B D, Christensen N J. Optimizing photon mapping using multiple photon maps for irradiance estimates[J]. 2003.

[31] Schjøth L, Sporring J, Fogh Olsen O. Diffusion based photon mapping[C]//Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2008, 27(8): 2114-2127.

[32] Jensen H W. Realistic image synthesis using photon mapping[M]. Natick: Ak Peters, 2001.

[33] Tabellion E, Lamorlette A. An approximate global illumination system for computer generated films[M]//ACM SIGGRAPH 2004 Papers. 2004: 469-476.

[34] Chen S E, Rushmeier H E, Miller G, et al. A progressive multi-pass method for global illumination[C]//Proceedings of the 18th annual conference on Computer graphics and interactive techniques. 1991: 165-174.

[35] Krivanek J, Gautron P. Practical global illumination with irradiance caching[J]. Synthesis lectures on computer graphics and animation, 2009, 4(1): 1-148.

[36] Jones N L, Reinhart C F. Irradiance caching for global illumination calculation on graphics hardware[J]. Ashrae/Simbuild, 2014: 111-20.

[37] Kaplanyan A, Dachsbacher C. Cascaded light propagation volumes for real-time indirect illumination [C]//Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games. 2010: 99-107.

[38] Yao C, Wang B, Chan B, et al. Multi-image based photon tracing for interactive global illumination of dynamic scenes [C]//Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2010, 29(4): 1315-1324.

[39] Jönsson D, Ynnerman A. Correlated photon mapping for interactive global illumination of time-varying volumetric data [J]. IEEE transactions on visualization and computer graphics, 2016, 23(1): 901-910.

[40] Kán P, Kaufmann H. Differential irradiance caching for fast high-quality light transport between virtual and real worlds [C]//2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE, 2013: 133-141.

[41] Lavignotte F, Paulin M. Scalable photon splatting for global illumination[C]//Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia. 2003: 203-ff.

[42] Kang C, WAN L, Xu Y, et al. A survey of photon mapping state-of-the-art research and future challenges [J]. Frontiers of Information Technology & Electronic Engineering, 2016, 17(3): 185-199.

[43] Jensen H W, Christensen N J. A practical guide to global illumination using photon maps [J]. SIGGRAPH 2000 Course Notes CD-ROM, 2000.

[44] Jensen H W, Christensen P H, Kato T, et al. A practical guide to global illumination using photon mapping [J]. SIGGRAPH 2002 Course Notes CD-ROM, 2002.

[45] Jensen H W. A practical guide to global illumination using ray tracing and photon mapping[M]//ACM SIGGRAPH 2004 Course Notes. 2004: 20-es.

[46] Zeng Z, Wang L, Wang B B, et al. Denoising stochastic progressive photon mapping renderings using a multi-residual network [J]. Journal of Computer Science and Technology, 2020, 35: 506-521.

[47] Tabellion E, Lamorlette A. An approximate global illumination system for computer generated films [M]// ACM SIGGRAPH 2004 Papers. 2004: 469-476.

[48] Dutre P, Bekaert P, Bala K. Advanced global illumination[M]. CRC Press, 2018.

[49] Hey H, Purgathofer W. Advanced radiance estimation for photon map global illumination[C]//Computer Graphics Forum. Oxford, UK: Blackwell Publishing, Inc, 2002, 21(3): 541-545.

[50] Havran V, Herzog R, Seidel H P, et al. Fast final gathering via reverse photon mapping[C]//Untitled Event. Blackwell, 2005: 323-333.

[51] Schregle R. Bias compensation for photon maps[C]//Computer Graphics Forum. Oxford, UK and Boston, USA: Blackwell Publishing, Inc, 2003, 22(4): 729-742.

[52] Purcell T J, Donner C, Cammarano M, et al. Photon mapping on programmable graphics hardware[M]// ACM SIGGRAPH 2005 Courses. 2005: 258-es.

[53] Suykens F, Willems Y D. Density control for photon maps[C]//Rendering Techniques 2000: Proceedings of the Eurographics Workshop in Brno, Czech Republic, June 26–28, 2000 11. Springer Vienna, 2000: 23-34.

[54] Jensen H W, Christensen P H. Efficient simulation of light transport in scenes with participating media using photon maps [C]//Proceedings of the 25th annual conference on Computer graphics and interactive techniques. 1998: 311-320.

[55] Moon J T, Marschner S R. Simulating multiple scattering in hair using a photon mapping approach[J]. ACM Transactions on Graphics (TOG), 2006, 25(3): 1067-1074.

[56] Jensen H W. Importance driven path tracing using the photon map[C]//Rendering Techniques' 95: Proceedings of the Eurographics Workshop in Dublin, Ireland, June 12–14, 1995 6. Springer Vienna, 1995: 326-335.

[57] Jensen H W, Christensen N J. Efficiently rendering shadows using the photon map[C]//Edugraphics+ Compugraphics Proceedings. 1995: 285-291.

[58] Jensen H W, Christensen N J. Photon maps in bidirectional Monte Carlo ray tracing of complex objects[J]. Computers & Graphics, 1995, 19(2): 215-224.

[59] Berndt R, Gimzewski J K. Photon emission in scanning tunneling microscopy: Interpretation of photon maps of metallic systems [J]. Physical Review B, 1993, 48(7): 4746.

[60] Fabianowski B, Dingliana J. Interactive global photon mapping[C]//Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2009, 28(4): 1151-1159.

[61] McGuire M, Luebke D. Hardware-accelerated global illumination by image space photon mapping[C]// Proceedings of the Conference on High Performance Graphics 2009. 2009: 77-89.

[62] Kang C, WAN L, Xu Y, et al. A survey of photon mapping state-of-the-art research and future challenges [J]. Frontiers of Information Technology & Electronic Engineering, 2016, 17(3): 185-199.

[63] Jensen H W, Christensen P. High quality rendering using ray tracing and photon mapping[M]//ACM SIGGRAPH 2007 courses. 2007: 1-es.

[64] Günther J, Wald I, Slusallek P. Realtime Caustics using Distributed Photon Mapping[J]. Rendering Techniques, 2004, 4(111-122): 3.

[65] Larsen B D, Christensen N J. Simulating Photon Mapping for Real-time Applications[J]. Rendering Techniques, 2004, 4: 123-132.

[66] Schregle R, Grobe L, Wittkopf S. Progressive photon mapping for daylight redirecting components[J]. Solar Energy, 2015, 114: 327-336.

[67] Zhu S, Xu Z, Jensen H W, et al. Deep kernel density estimation for photon mapping[C]//Computer Graphics Forum. 2020, 39(4): 35-45.

[68] Frisvad J R, Schjøth L, Erleben K, et al. Photon differential splatting for rendering caustics[C]//Computer Graphics Forum. 2014, 33(6): 252-263.

[69] Jonsson D, Ynnerman A. Correlated Photon Mapping for Interactive Global Illumination of Time-Varying Volumetric Data [J]. IEEE Transactions on Visualization & Computer Graphics, 2016, 23(1): 901-910. DOI:10.1109/TVCG.2016.2598430.

[70] Evangelou I, Papaioannou G, Vardis K, et al. Rasterisation-based progressive photon mapping[J]. The Visual Computer, 2020, 36: 1993-2004.

[71] Hurter C, Ersoy O, Telea A. Graph bundling by kernel density estimation[C]//Computer graphics forum. Oxford, UK: Blackwell Publishing Ltd, 2012, 31(3pt1): 865-874.

[72] Ritschel T, Engelhardt T, Grosch T, et al. Micro-rendering for scalable, parallel final gathering[J]. ACM Transactions on Graphics (TOG), 2009, 28(5): 1-8.

[73] Hachisuka T. Final gathering on GPU[C]//ACM Workshop on General Purpose Computing on Graphics Processors. 2004, 3.

[74] Tawara T, Myszkowski K, Seidel H P. Localizing the Final Gathering for Dynamic Scenes using the Photon Map[C]//VMV. 2002: 69-46.

[75] Czuczor S, Szirmay-Kalos L, Szécsi L, et al. Photon Map Gathering on the GPU[C]//Eurographics (Short Presentations). 2005: 117-120.

[76] Zeng Z, Wang L, Wang B B, et al. Denoising stochastic progressive photon mapping renderings using a multi-residual network[J]. Journal of Computer Science and Technology, 2020, 35: 506-521.

[77] Günther T, Grosch T. Distributed out-of-core stochastic progressive photon mapping[C]//Computer Graphics Forum. 2014, 33(6): 154-166.

[78] Weiss M, Grosch T. Stochastic progressive photon mapping for dynamic scenes[C]//Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2012, 31(2pt3): 719-726.

[79] Belcour L, Soler C. Frequency based kernel estimation for progressive photon mapping[M]//SIGGRAPH Asia 2011 Posters. 2011: 1-1.

[80] Zhdanov A D, Zhdanov D D. Progressive backward photon mapping[J]. Programming and Computer Software, 2021, 47(3): 185-193.

[81] Pedersen S A. Progressive photon mapping on gpus[D]. Institutt for datateknikk og informasjonsvitenskap, 2013.

[82] Wallace J R, Cohen M F, Greenberg D P. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods[J]. ACM Siggraph Computer Graphics, 1987, 21(4): 311-320.

[83] Dimov I T, Penzov A A, Stoilova S S. Parallel Monte Carlo approach for integration of the rendering equation[C]//Numerical Methods and Applications: 6th International Conference, NMA 2006, Borovets, Bulgaria, August 20-24, 2006. Revised Papers 6. Springer Berlin Heidelberg, 2007: 140-147.

[84] Zimmerman K. Developing the rendering equations[J]. A Basic. Indiana University, 1998.

[85] Yu T T, Lowther J, Shene C K. Photon mapping made easy[C]//Proceedings of the 36th SIGCSE technical symposium on Computer science education. 2005: 201-205.

[86] Schregle R. Daylight simulation with photon maps[J]. 2004.