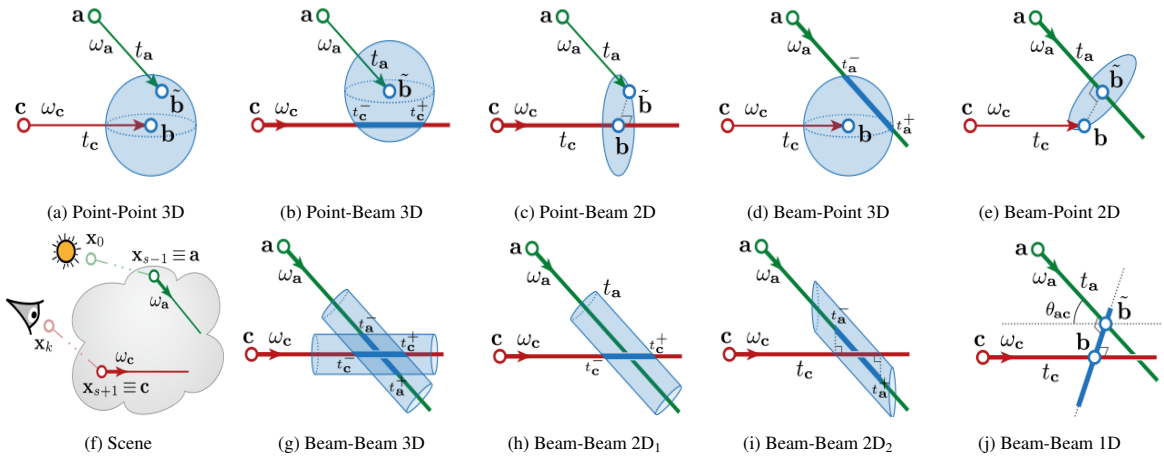
Global Illumination in Participating Media

Adam Papp*
Vienna University of Technology



(a) Point-Point 3D (b) Point-Beam 3D (c) Point-Beam 2D (d) Beam-Point 3D (e) Beam-Point 2D

(f) Scene (g) Beam-Beam 3D (h) Beam-Beam $2D_1$ (i) Beam-Beam $2D_2$ (j) Beam-Beam 1D

Figure 1: Illustration of different radiance estimators described by [Jarosz et al. 2011a], where 1a illustrates photon mapping, 1e illustrates beam radiance estimation and 1j illustrates photon beams. As mentioned by [Jarosz et al. 2011a] photon beams can be seen as a generalization of virtual point lights. This image was taken from [Křivánek et al. 2014]

Abstract

This report provides a survey of several state-of-the-art algorithms to render global illumination in participating media. Photon mapping, radiance estimation methods, virtual light methods and a technique to model photon density using Gaussian mixtures are described. The description of the algorithms focuses on giving a brief overview of the radiance evaluation and properties like biased or unbiased, handling heterogeneous media and anisotropic scattering.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

Keywords: global illumination, participating media, survey

1 Introduction

Photorealistic rendering is an important task in many applications. In the every day life humans have the most interaction with it in entertainment, for example movies, games or virtual reality. It has also an impact on virtual simulations and safety design, e.g. design of emergency exit signs and simulation of a room filled with smoke. Global illumination algorithms consider not just the direct incoming light (e.g. Phong reflection model [Phong 1975]) from the lightsource, but also the reflected light from other objects. This is often costly to compute and there is a trade-off between quality of the rendered image and rendering time. Some of these algorithms consider the light traveling through vacuum, however in reality, the light is traveling through a medium. In order to achieve photorealistic images, participating media (e.g. fog, smoke or water) must be considered. An extensive survey was written by [Cerezo et al. 2005] about global illumination in participating media. Many algorithms have been developed to achieve global illumination in real-time, but either they take constraints of the scene (e.g. underwater texture or analytic solutions tied to fog rendering, as mentioned by [Cerezo et al. 2005]) or they render not physically correct or in other words near-photorealistic solutions. An example is the ambient occlusion method, which is often used in computer games or molecule visualization [Tarini et al. 2006].

This paper focuses on offline rendering algorithms that consider general participating media. Section 2 discusses the theoretical solutions and problems of rendering in participating media. The sections after are describing briefly the idea of selected methods and their improvements. They are sorted in ascending order of time of publication as; photon mapping (section 3), radiance caching (section 4), beam radiance estimate (section 5), expectation-maximization for photon mapping (section 6), photon beams (section 7) and virtual beam light method (section 8). The final section 9 contains a brief summary and results of the above mentioned methods and an outlook of future approaches.

*e-mail: e1327381@student.tuwien.ac.at

2 Preliminaries

In this section mathematical equations are presented that are used in global illumination. First, the rendering equation is discussed. Second, the radiative transfer equation is detailed, which describes the light behavior in participating media.

2.1 The Rendering Equation

A general model for light transport is given by the rendering equation [Kajiya 1986]. The rendering equation (1) describes the amount of radiance leaving a point $\mathbf{x}$ into the direction $\vec{\omega}$,

$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_\Omega f_r(\mathbf{x}, \vec{\omega_i}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega_i}) (\vec{\omega_i} \cdot \mathbf{n}) \, d\vec{\omega_i}, \tag{1}$$

where the notations are defined in figure 2.

Symbol	Description
$\mathbf{x}$	location in space
$\mathbf{n}$	surface normal at location $\mathbf{x}$
$\vec{\omega}$	direction of the outgoing light
$\vec{\omega_i}$	direction of the incoming light
Ω	the unit hemisphere centered around $\mathbf{n}$ containing all possible values for $\vec{\omega_i}$
$\vec{\omega_i} \cdot \mathbf{n}$	weakening factor of inward irradiance due to incident angle or light attenuation
$f_r(\mathbf{x}, \vec{\omega_i}, \vec{\omega})$	*bidirectional reflectance distribution function* (BRDF), the proportion of light reflected from $\vec{\omega_i}$ to $\vec{\omega}$ at position $\mathbf{x}$
$L_e(\mathbf{x}, \vec{\omega})$	emitted spectral radiance
$L_i(\mathbf{x}, \vec{\omega_i})$	spectral radiance toward $\mathbf{x}$ from direction $\vec{\omega_i}$
$L(\mathbf{x}, \vec{\omega})$	total spectral radiance along direction $\vec{\omega}$ from a particular position $\mathbf{x}$

Figure 2: Notations of the rendering equation

Since the rendering equation must be solved numerically, the quality of the rendered image depends on the number of samples. Therefore an algorithm that solves the rendering equation can be characterized by consistence and bias. An algorithm is consistent if it converge to the exact solution after an infinite amount of time. An algorithm is unbiased if the expected error is zero, regardless of the number of samples. An intuition of a biased algorithm is, that rendering more does not guarantee that the result will not be worse. Figure 3 illustrates three error functions, where consistence and bias is described. One of the misconceptions of bias is, pointed out by [Hachisuka 2013], that unbiased means more accurate, however biased algorithm might produce better result in an iteration and converge faster compared to an unbiased algorithm. The consistency (2) and bias (3) are defined as

$$\lim_{N \to \text{inf}} E_N[F] = \int f(x) \, dx, \tag{2}$$

$$E_N[F - \int f(x) \, dx] = 0. \tag{3}$$

2.2 Radiative Transfer Equation

The Radiative Transfer Equation (RTE) by [Chandrasekhar 1960] describes the process, when photons in participating media might alter their path, get scattered or absorbed and reduce strength in the original direction. These interactions are visualized on figure 5. One of the limitation of the rendering equation, that it does not consider participating media. It assumes, that the photons travel unobstructed until they interact with a surface and it does not take

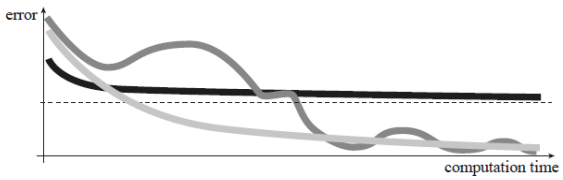


Figure 3: Example of three error functions. The black error function is from a not consistent algorithm, since it does not converge to zero, both gray functions are from consistent ones. The dark gray function is from a biased algorithm, because rendering more does not guarantee that the result will not be worse. The light gray and black functions are from an unbiased algorithm. This image was taken from http://www.cs.columbia.edu/~keenan/Projects/Other/BiasInRendering.pdf, which was visited on 06/12/2015.

into account transmission, absorption and scattering. The RTE (4) describes also the amount of radiance leaving a point $\mathbf{x}$ into the direction $\vec{\omega}$, while considering all the interactions mentioned before:

$$L(\mathbf{x}, \vec{\omega}) = T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L_s(\mathbf{x}_s, \vec{\omega}) + L_m(\mathbf{x}, \vec{\omega}), \tag{4}$$

where the medium radiance L_m is computed as:

$$L_m(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega}) \, dt, \tag{5}$$

surface radiance L_s is defined by the rendering equation and other notations are defined in figure 4.

Symbol	Description
$\mathbf{x}$	location in space
$\mathbf{x}_s$	nearest surface along the ray at $\mathbf{x} - s\vec{\omega}$
$\vec{\omega}$	direction of the outgoing light
$\sigma_s(\mathbf{x})$	Scattering coefficient at $\mathbf{x}$
$\sigma_a(\mathbf{x})$	Absorption coefficient at $\mathbf{x}$
$\sigma_t(\mathbf{x})$	Extinction coefficient at $\mathbf{x}$
$p(\mathbf{x}, \vec{\omega}, \vec{\omega}')$	Normalized phase function
$\tau(\mathbf{x} \leftrightarrow \mathbf{x}')$	Optical thickness: $\int_{\mathbf{x}'}^{\mathbf{x}} \sigma_t(x) \, dx$
$T_r(\mathbf{x} \leftrightarrow \mathbf{x}')$	Transmittance: $e^{-\tau(x \leftrightarrow x')}$
$L(\mathbf{x}, \vec{\omega})$	Incident radiance at $\mathbf{x}$ from $\vec{\omega}$
$L_i(\mathbf{x}, \omega_i)$	spectral radiance toward $\mathbf{x}$ from direction ω_i

Figure 4: Notations of the radiative transfer equation

Algorithms often distinguish between single scattering and multiple scattering. Single scattering is when the radiance undergoes a single scattering event along it's path from a surface to the eye. Multiple scattering when the radiance is at least once scattered in the medium before it reaches $\mathbf{x}$. Multiple scattering is more expensive then single scattering. Multiple scattering can be ignored, when the participating medium is thin or has a low albedo. If the participating media does not scatter, e.g. rendering fire (mentioned in [Cerezo et al. 2005]), the scattering can be omitted from the computation.

When the scattering in a medium happens uniformly in every direction, then the media is called isotropic. Anisotropic scattering is when the scattering is depending on the phase function. The medium can be described as homogeneous, when the $\sigma_s(\mathbf{x})$, $\sigma_a(\mathbf{x})$ and $\sigma_t(\mathbf{x})$ is constant in all $\mathbf{x}$ within the medium. In a heterogeneous media, these values vary within different $\mathbf{x}$.

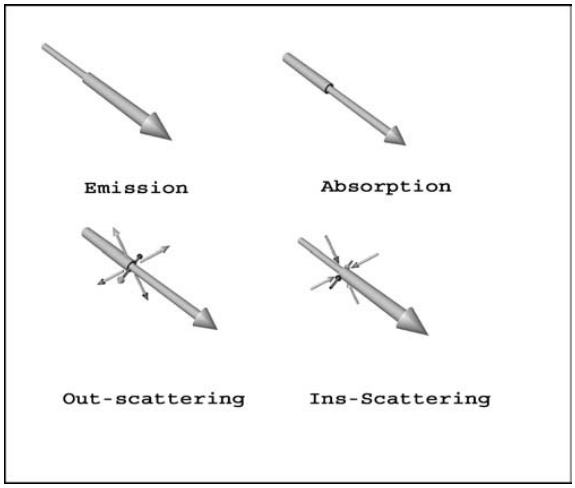


Figure 5: Visualization of light interaction in participating medium. This image was taken from [Cerezo et al. 2005]

3 Photon Mapping

Volumetric photon mapping is a global illumination algorithm by [Jensen and Christensen 1998], which does not make additional assumption of the medium properties that is being rendered. It can handle isotropic, anisotropic, homogeneous and heterogeneous media of arbitrary albedo. Photon mapping is a biased algorithm. The RTE (4) is solved by a combination of photon tracing, ray marching and density estimation. At first, photons are shot from the light sources. This is usually done using Markov Random Walk, but other sampling solutions can be used as mentioned by [Jarosz et al. 2011a]. Photons are scattered at surfaces and withing the medium and their interactions are stored in a global data structure, e.g. kd-tree by [Jarosz et al. 2008b]. In the second step, ray marching is used to solve numerically the RTE (4) as:

$$L(\mathbf{x},\vec{\omega}) \approx T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L_s(\mathbf{x}_s,\vec{\omega}) + \sum_0^s T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\,\Delta t \tag{6}$$

In equation (6), the in-scattered radiance L_i is the most expensive to compute, because at point x_t all arriving light from other points have to be considered. Using photon mapping it is possible to approximate the radiance by efficiently reusing the computation performed in the photon tracing state, by gathering nearby photons from point x_t within a spherical neighborhood of radius r,

$$L_i(\mathbf{x},\vec{\omega}) \approx \sum_{i=1}^{n} \frac{p(\mathbf{x}_t,\vec{\omega},\vec{\omega_i})\Delta\phi_i}{\frac{4}{3}\pi r^3} \tag{7}$$

where $\Delta\Phi_i$ is the power of photon i, and $\vec{\omega_i}$ is its incident direction [Jensen and Christensen 1998].

[Jensen and Christensen 1998] approximates the in-scattered radiance L_i at fixed points within the scene. To obtain a useful estimate of radiance at all points in the scene a constant three-dimensional kernel with a radius based on the n^{th} nearest neighbor is used. This blurring effect is the reason for the bias in the photon mapping method.

Algorithm 1 Photon mapping

```
1:  while stopping criteria not met do
2:      start photon tracing p from light source
3:      trace photon p
4:      if p hits surface s or inside medium m then
5:          if s is diffuse or inside m then
6:              store p in photon map
7:          end if
8:          if s is not diffuse or inside m then
9:              reflect, transmit, absorb, scatter
10:             go to 3
11:         end if
12:     end if
13:     for all pixel p do                          ▷ rendering
14:         ray tracing from p to camera until surface is hit
15:         density estimation
16:     end for
17: end while
```

3.1 Progressive Photon Mapping

[Hachisuka et al. 2008] improved the photon mapping algorithm by decreasing the required memory usage. When a scene has a large extent, many photons would be required to render, which is a bottleneck of the photon mapping.

Progressive photon mapping (PPM) is a multipass algorithm, where in the first pass, using ray tracing, all the surfaces are found in the scene, which are visible through each pixel. These points on the surfaces are called hitpoints. All subsequent passes are photon tracing passes. In one photon tracing pass photons are shot from the lightsource and the estimate for each pixel is refined. The algorithm is not limited by memory as the photons are discarded after each iteration.

Progressive radiance estimate merges the result from several photon tracing passes. First the radius $R(x)$ is reduced due to the assumption that photon density is constant and increased number of photons. Then the unnormalized flux Φ'_p is adjusted to take into account the reduced radius. The radiance is evaluated as:

$$L(\mathbf{x},\vec{\omega}) \approx \frac{1}{\pi R(x)^2} \frac{\sum_{p=1}^{N_{hitpoint}} f_r(\mathbf{x},\vec{\omega},\vec{\omega_p})\Delta\Phi_p(\mathbf{x}_p,\vec{\omega_p})}{N_{photons}} \tag{8}$$

4 Radiance Caching

[Jarosz et al. 2008a] presented a similar approach in spirit to volumetric photon mapping in that illumination information is stored and reused in a volume. The in-scattered radiance L_i in equation (6) is computed using Monte Carlo ray tracing based on a combination of ray marching and random walk sampling. Based on the assumption that the distribution of in-scattered radiance is often smooth in large parts of the participating medium, radiance is cached at a sparse set of locations using an octree. To evaluate the radiance at nearby points extrapolation is used and to improve it's accuracy the gradient of the cached point is stored. Figure 6 visualizes the computation of single and multiple scattering radiance and gradient.

In single scattering cases, transmission for homogeneous media can be computed explicitly. For heterogeneous media ray marching is used with a fixed step size and single random offset. The distribution of the outgoing direction $\vec{\omega}$ in isotropic media is uniform. In

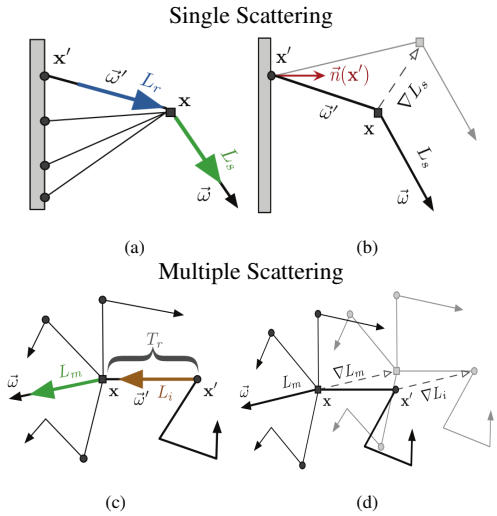


Figure 6: Visualization of gradient computation by [Jarosz et al. 2008a]. In single scattering the gradient ∇L_s is computed by translation w.r.t the evaluation point $\mathbf{x}$. In multiple scattering, after the distribution of points by random walk-paths, the whole path is translated as one. This image was taken from [Jarosz et al. 2008a]

anisotropic media the outgoing direction $\vec{\omega}$ depends on the shape of the phase function. A compact representation of the function using spherical harmonic expansion is computed. Spherical harmonics are used to represent functions defined on the surface of a sphere by a series of functions. Multiple scattering is computed using Monte Carlo path tracing for both homogeneous and heterogeneous media. Although the radiance in isotropic and anisotropic media can be computed the same way as in single scattering, the gradient in anisotropic media has to be computed using importance sampling.

[Jarosz et al. 2008a] described, that the intensity of the light in participating media falls off exponentially, therefore the in-scattered radiance is exponentially extrapolated from the cached values. Linear extrapolation in log-space is equivalent to exponential extrapolation. The extrapolated radiance estimate defined by [Jarosz et al. 2008a] approximates the radiance at point $\mathbf{x}$ from a set of cache points C whose valid radii contain the query location by a weighted sum of extrapolated radiance values,

$$L(\mathbf{x}) \approx exp\left(\frac{\sum_{k\in C}\left(ln(L_k)+\frac{\nabla L_k}{L_k}\cdot(\mathbf{x}-\mathbf{x}_k)w(d_k)\right)}{\sum_{k\in C} w(d_k)}\right) \tag{9}$$

where L_k, ∇L_k, $\mathbf{x}_k$ and r_k are the radiance, gradient, position and valid radius of cache point k respectively. The weighting function w is a smooth cubic, $w(d) = 3d^2 - 2d^3$. The valued_k is defined as $d_k = 1 - ||\mathbf{x}_k - \mathbf{x}||/r_k$ so that the weighting function has the most influence at the cache location, and this influence falls off smoothly to zero at the valid radius. A comparison of exponential extrapolation to other methods can be seen of figure 7.

5 Beam Radiance Estimate

[Jarosz et al. 2008b] pointed out that equation (7) used in photon mapping is suboptimal, because photons may contribute more then

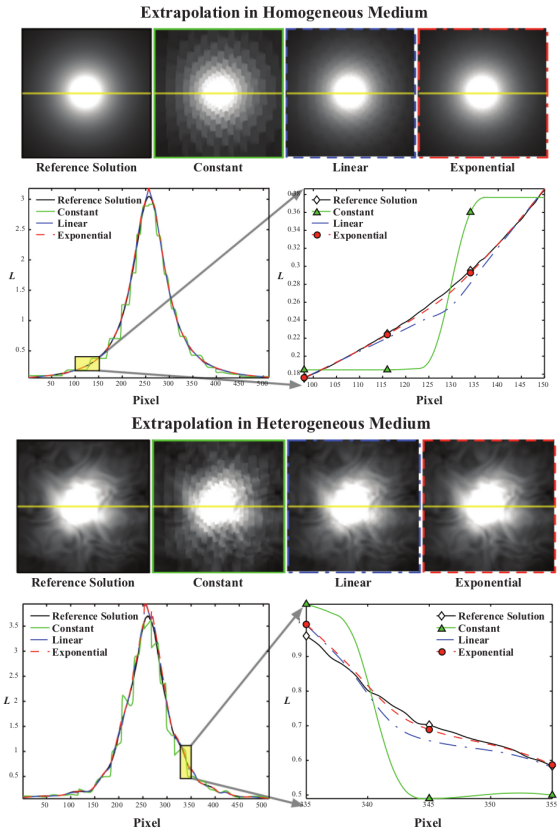


Figure 7: A comparison by [Jarosz et al. 2008a] of extrapolation methods. The visualization shows that both in homogeneous (up) and heterogeneous (down) medium the exponential extrapolation proposed by [Jarosz et al. 2008a] reconstructs the sample function more accurately. This image was taken from [Jarosz et al. 2008a]

once in the gathering process or if the step size is too large then photons might be omitted. Compared to conventional gathering, beam gathering queries all photons once along the entire ray. Figure 8 shows the difference between conventional gathering and beam gathering.

[Jarosz et al. 2008b] replaces the photon marching step by using the beam radiance estimate to accumulate the in-scattered radiance along an entire ray. The kD-tree photon storage is extended into a bounding box hierarchy and photon discs are stored. The probability of photons reaching exactly a ray is zero, therefore a blur kernel is applied to each photon to blur a photon and generate a disc. The radius of each disc depends on the local density of the photon. This blurring procedure introduces the bias in the algorithm. The blurring method is visualized on figure 9. The photon discs are found along viewing rays by traversing the hierarchy and locating all photons whose bounding spheres intersect the ray. The beam radiance estimate is defined as

$$L_m(\mathbf{x},\vec{\omega},s) \approx \frac{1}{N}\sum_{i=1}^{N} K_i(\mathbf{x},\vec{\omega},s,\mathbf{x}_i,r_i)T_r(\mathbf{x}\leftrightarrow\mathbf{x}'_i)\sigma_s(\mathbf{x}'_i)p(\mathbf{x}_i,\vec{\omega},\vec{\omega}_i)\alpha_i \tag{10}$$

where $\mathbf{x}'_i = \mathbf{x} + t_i\vec{\omega}$ is the projection of the photon location $\mathbf{x}_i$ onto the ray's direction $\vec{\omega}$, and $t_i = (\mathbf{x}_i - \mathbf{x})\cdot\vec{\omega}$ and K_i is the blurring kernel. The kernel is optimized for two dimensions since the radiance

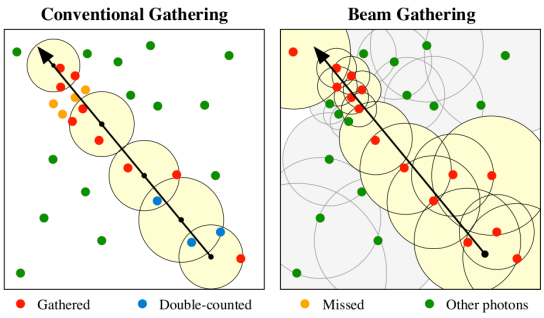


Figure 8: The method by [Jarosz et al. 2008b] (right) assigns a radius to each photon and finds all photons along the length of an entire ray. Conventional gathering (left) searches in distinct positions along the ray and might double-count or miss photons. This image was taken from [Jarosz et al. 2008b]

already includes the integration along the ray itself.

The transmission $T_r(\mathbf{x} \leftrightarrow \mathbf{x}'_i)$ in homogeneous media can be computed explicitly during gathering. In heterogeneous media, before the gathering process, with ray marching a lookup table is built, where transmission values are stored. At the gathering stage, the transmission is computed by interpolating within the lookup table.

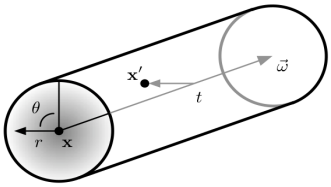


Figure 9: Visualization of the blurring kernel in the beam radiance estimate [Jarosz et al. 2008b]. The photon coordinate $\mathbf{x}'$ is parameterized in cylindrical coordinates, (t, θ, r), about the ray $(\mathbf{x}, \vec{\omega})$. This image was taken from [Jarosz et al. 2008b]

6 Progressive Expectation-Maximization

The approach by [Jakob et al. 2011] is fitting anisotropic Gaussian mixtures to the photon density and radiance distribution, providing faster rendering and reduced variance. Their method replaces photons with multivariate Gaussians, which can be seen as the replacement of a non-parametric density estimate with a parametric density model.

Their solution is addressing the bottlenecks of the beam radiance estimate, when high-frequency lighting requires an excessive number of photons and when the scene has a large extent and many photons would be processed, which contributes only a little to the final result. The advantage over beam radiance estimate is shown on figure 10

There are number of similarities between their solution and to hierarchical photon mapping by [Spencer and Jones 2009a], which provides a level-of-detail for gathering, and photon relaxation by [Spencer and Jones 2009b], which reduces variance by updating positions of photons iteratively. The Gaussian mixtures are constructed in a hierarchical representation and stored in a bounding box hierarchy. When tracing a ray, entire mixtures can be discarded.

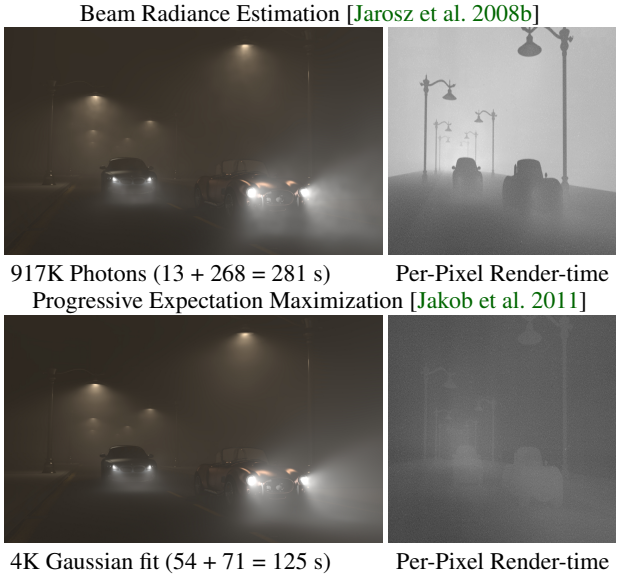


Figure 10: The beam radiance estimate by [Jarosz et al. 2008b] (up) renders slower the CARS scene compared to the Gaussian mixture model fitting by [Jakob et al. 2011] (down), due to the scene extent. The complete rendering time is summed by the preprocessing stage and the rendering state, respectively. This image was taken from [Jakob et al. 2011]

The mixtures are updated progressively by shooting out new photons if the stored statistics of the mixture is too noisy. They call this method as the Progressive Expectation-Maximization.

[Jakob et al. 2011] approximates the radiance by evaluating the contribution of each Gaussian using integration along $\mathbf{r}(t) = \mathbf{x} + t\vec{\omega}$ and accounting for transmittance,

$$L_m^j(\mathbf{x}, \vec{\omega}, b) = \frac{1}{4\pi} \overline{w_j} \int_0^b g(\mathbf{r}(t) | \overline{\Theta_j}) T_r(\mathbf{x} \leftrightarrow \mathbf{r}(t))\, dt \qquad (11)$$

where j is the index into GMM pyramid nodes, Θ_j are the parameters(weight, mean, covariance) of the GMM pyramid component and $\overline{w}$ is the weight. In homogeneous media the integral can be solved in close-form solution. In heterogeneous media, the integral could be computed using ray marching. The discussed implementation by [Jakob et al. 2011] handles only isotropic media, although it is believed by the authors, that it could be extended for anisotropic media by additional storage of directional information during fitting.

7 Progressive Photon Beams

The photon beam approach by [Jarosz et al. 2011a] treats each photon as a beam of light starting at the photon positions and shooting in the photon's outgoing direction. It is a generalization of photon mapping and accelerates rendering in participating media by performing density estimation on the full paths of photons, instead of just photon scattering locations. The incident radiance is estimated for one photon beam as:

$$L_m(\mathbf{x}, \vec{\mathbf{w}}, r) \approx k_r(u)\sigma_s(\mathbf{x}_w)\Phi T_r(w) T_r(v) \frac{f(\vec{\mathbf{w}} \cdot \vec{\mathbf{v}})}{sin(\vec{\mathbf{w}}, \vec{\mathbf{v}})} \qquad (12)$$

where k_r is the blurring kernel and Φ is the power of the photon. $T_r(w)$ accounts for attenuation through distance w to $\mathbf{x}$ and $T_r(v)$ computes the transmittance through distance v to the start of the beam. The coordinate system $(\vec{\mathbf{u}}, \vec{\mathbf{v}}, \vec{\mathbf{w}})$ is defined as $\vec{\mathbf{w}}$ the query ray direction, $\vec{\mathbf{v}}$ the direction of the photon beam and $\vec{\mathbf{u}} = \vec{\mathbf{v}} \times \vec{\mathbf{w}}$. The scalars (u, v, w) are signed distances along the three axes to the imaginary photon point closest to the query ray. The photon beam is treated as an infinite number of imaginary photon points along its length. Figure 11 visualizes this concept.

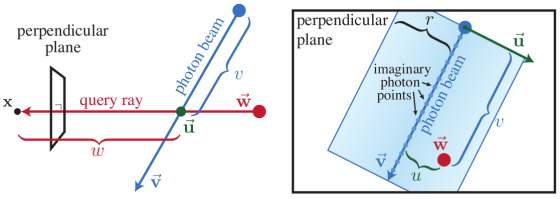


Figure 11: Visualization of a photon beam viewed from the side (left) and the plane perpendicular to the query ray (right). This image was taken from [Jarosz et al. 2011b]

The photon beams method is biased, since beams are blurred with a finite width in 1D. Applying photon beams to progressive photon mapping is not straightforward, due to fundamental differences between photon beams and photon points, furthermore photon beams are formulated using variable kernel density estimation. Progressive photon beams by [Jarosz et al. 2011b] approximates the medium radiance by Monte Carlo estimation of many photon beams as

$$\overline{C}_N = \frac{1}{N} \sum_{i=1}^{N} \frac{W(\mathbf{x}_i, \vec{\mathbf{w}}_i) L_m(\mathbf{x}_i, \vec{\mathbf{w}}_i)}{p(\mathbf{x}_i, \vec{\mathbf{w}}_i)} \tag{13}$$

where $\overline{C}_N$ is the pixel intensity, W is a function that weights the contribution of L_m and $p(\mathbf{x}_i, \vec{\mathbf{w}}_i)$ denotes the probability density of generating a particular position and direction when tracing paths from the eye. Two steps are repeated, where the first pass is similar to photon mapping's photon tracing, and appropriate beam widths and kernel widths from photon differentials are computed in addition. In the second pass, radiance along each ray is computed using equation (12). In homogeneous media this can be solved in close form. The transmittance T_r in heterogeneous media is solved using progressive deep shadow maps, which is an unbiased estimator for the transmittance. Anisotropic scattering is possible with photon beams.

8 Virtual Light Methods

Virtual point light method introduced by [Keller 1997] converts each photon into a "virtual" point light source within a traditional photon map. This concept allows to simplify multiple scattering to standard direct lighting technique. The in-scattered radiance of a single virtual point light is computed as:

$$L_i \approx \frac{\Phi f_s(\theta_p) f_s(\theta_u) T_r(w) V}{w^2}, \tag{14}$$

where Φ is the power of the photon, w and V are the distance and binary visibility from $\mathbf{x}_u$ to the point light, $f_s(\theta_p)$ is the phase function at the point light and $f_s(\theta_u)$ accounts for scattering at the evaluation location. Virtual point light method suffers from artifacts that form spikes of high intensity. These can be resolved by clamping or blurring on the cost of significant bias.

8.1 Virtual Ray Lights

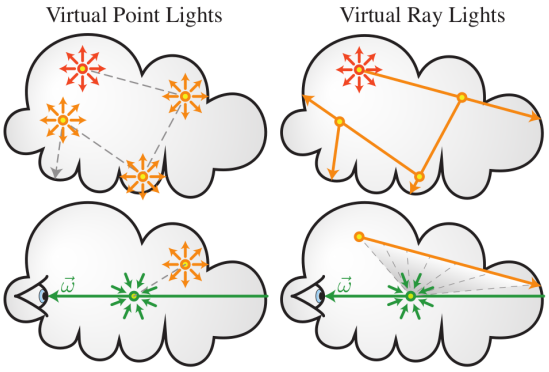


Figure 12: The virtual ray light method by [Novák et al. 2012b] (right) converts entire segments of the random-walk into virtual ray lights, compared to virtual point lights (left), where only the vertices are converted. This image was taken from [Novák et al. 2012b]

[Novák et al. 2012b] introduced the virtual ray light method, which estimates radiance using entire path segments created during photon tracing and computes an unbiased multiply scattered light from the media onto itself and surfaces. This is illustrated on figure 12. As this method is unbiased, progressive updating of results is trivial. The radiance equations in (4) and (5) are further decomposed to:

$$L_s = L_s^s + L_s^m + L_s^l \text{ and } L_m = L_m^s + L_m^m + L_m^l, \tag{15}$$

where subscripts denote the location and superscripts denote the source of radiance. L_s^l and L_m^l are radiance arriving directly from the light sources and their computation is trivial. The in-scattered radiance is computed using the entire path segment created during photon tracing and computed as:

$$L_i \approx \Phi \int_0^t \frac{\sigma_s(v) f_s(\theta_v) f_s(\theta_u) T_r(w_u(v)) T_r(v) V_u(v)}{w_u(v)^2} \, dv, \tag{16}$$

where t is the length of the ray light with the parameter v. V_u and w_u are the binary visibility and distance from $\mathbf{x}_u$ to the point v on the virtual ray light. The phase function $f_s(\theta_v)$ evaluates scattering at the virtual ray light connection θ_v, while $f_s(\theta_u)$ evaluates the camera connection θ_u. L_s^m is computed analogous to L_i, where the phase function $f_s(\theta_u)$ is replaced by the cosine wighted BRDF f_r:

$$L_s^m \approx \Phi \int_0^t \frac{\sigma_s(v) f_s(\theta_v) f_r T_r(w_u(v)) T_r(v) V_u(v)}{w_u(v)^2} \, dv. \tag{17}$$

The equation L_m^m is given by inserting equation (16) into equation (5), which is estimated as a double integral along the camera ray and virtual ray light:

$$L_m^m \approx \Phi \int_0^t \int_0^t \frac{\sigma_s(u)\sigma_s(v) f_s(\theta_u) f_s(\theta_v) T_r(u) T_r(v) T_r(w) V}{w(u,v)^2}\, dvdu. \tag{18}$$

Figure 13 illustrates the computation of term L_m^m. The integrands in L_s^m and L_m^m are estimated with unbiased Monte Carlo integration as:

$$L_s^m \approx \frac{1}{N}\sum_{i=1}^{N} \frac{g_s^m(v_i)}{pdf(v_i)} \text{ and } L_m^m \approx \frac{1}{N}\sum_{i=1}^{N} \frac{g_m^m(u_i,v_i)}{pdf(u_i,v_i)} \tag{19}$$

where $pdf(v_i)$ in L_s^m is the probability of choosing a location along a virtual ray light for a given surface point and $pdf(u_i,v_i)$ is the probability of choosing points on the camera ray and the virtual ray light. The major contribution of [Novák et al. 2012b] is the importance sampling method of these estimators according to the phase function and inverse-squared distance for both isotropic and anisotropic scattering. L_s^s and L_m^s are not represented by virtual ray lights and their endpoints are handled with other methods (e.g. virtual point lights).

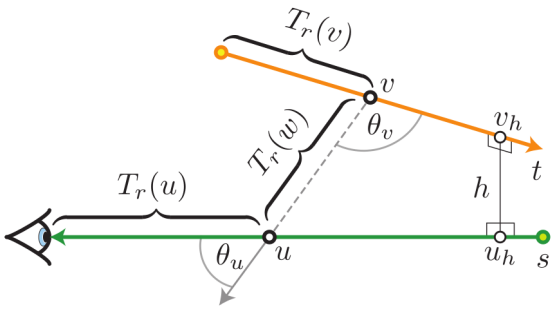


Figure 13: Illustration of equation (18), where the transport between an entire camera ray (green) and a virtual ray light (orange) is computed. This image was taken from [Novák et al. 2012b]

8.2 Progressive Virtual Beam Lights

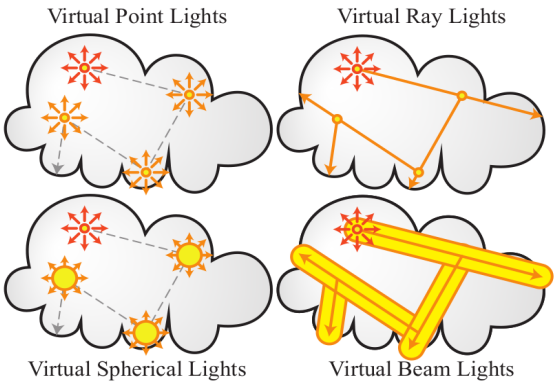


Figure 14: Virtual beam lights by [Novák et al. 2012a] was inspired by virtual spherical light. The virtual ray lights are expanded with finite thickness. This image was taken from [Novák et al. 2012a]

Virtual beam lights by [Novák et al. 2012a] further improves virtual ray light method. In images rendered using virtual ray light still many distracting singularities remain (especially for transport between surfaces and media where only a single, and not double, integration is performed), as mentioned by [Novák et al. 2012a]. To overcome this drawback, virtual ray lights are blurred into volumetric virtual beam lights. Figure 14 visualizes this concept. This blurring introduces bias, thus progressive update is solved based on the work of [Jarosz et al. 2011b], which minimizes the impact of the number of photons per pass on the final result.

The light from a surface to the medium L_s^m is obtained, by expanding each virtual ray light to a virtual spherical light, which is expressed as:

$$L_s^m \approx \int_0^t \sigma_s(v) T_r(v) T_r(s,v) L_s^{vsl}(s,v)\, dv, \tag{20}$$

where L_s^{vsl} integrates the scattering function over the solid angle of the virtual spherical light centered at v from the surface at s. The term L_s^{vsl} is solved using the beam radiance estimate [Jarosz et al. 2008b], which interprets the sphere as a disc, that faces the gather direction. The pdf of virtual ray light is modified to quickly approximate the maximum contribution of the integrand's term within the virtual spherical light. The term L_m^m is computed similarly by inflating virtual ray lights into virtual spherical light. It is computed as:

$$L_m^m \approx \int_0^s \int_0^t \sigma_s(u)\sigma_s(v) T_r(u) T_r(v) T_r(u,v) L_m^{vsl}(u,v)\, dvdu, \tag{21}$$

where $L_m^{vsl}(u,v)$ is defined analogously to $L_s^{vsl}(s,v)$ but using two phase function. First, using importance sampling by [Novák et al. 2012b], points v_i are taken along the virtual ray light and inflated to virtual spherical light. After the virtual spherical lights are obtained, L_m^m is computed analogously to L_m^s. Both L_m^s and L_s^s term is computed using virtual spherical light. The endpoint of a virtual ray light is inflated to a virtual spherical light and its contribution to the camera ray locations is evaluated numerically. The computation of these terms are visualized on figure 15.

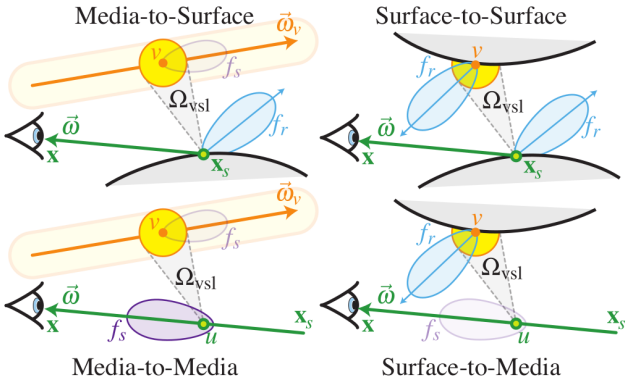


Figure 15: Visualization of each transport type computed by virtual beam light method. This image was taken from [Novák et al. 2012a]

9 Conclusion

Most of the algorithms presented in this survey precompute a photon map or use a similar photon storage mechanism. The main difference between these methods is the handling of the photons

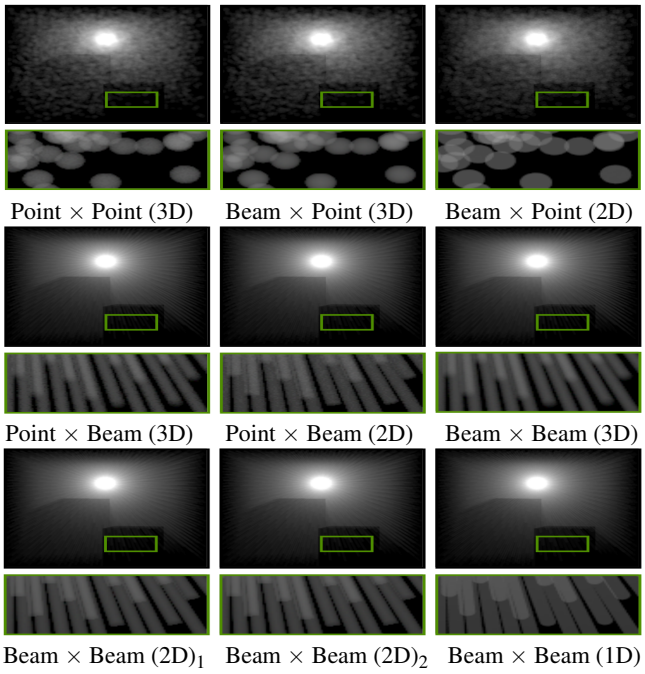


Figure 16: Experiment by [Jarosz et al. 2011a], showing the artifacts and blurring behaviors of radiance estimators. This image was taken from [Jarosz et al. 2011a]

contribution to the pixel values. Since accumulating each photons contribution to a pixel without blurring converges slowly, the photons are blurred along rays or beams, which introduces bias. Figure 1 illustrates the different blurring techniques, where figure 1a illustrates photon mapping by [Jensen and Christensen 1998], figure 1e illustrates the beam radiance estimate by [Jarosz et al. 2008b] and figure 1j illustrates photon beams by [Jarosz et al. 2011a]. Figure 16 illustrates the quality of these blurring techniques. Virtual beam light technique handle the photons as virtual lightsources and solves the blurring with the beam radiance estimate [Jarosz et al. 2008b]. The method by [Jakob et al. 2011] is an exception, because instead of a photon map, they use Gaussian mixtures to model the distribution of photons.

The increasing number of photons, increases the memory usage of an algorithm. Memory usage can be bounded by updating the result progressively and in each step storing only a predefined amount of photons. Since most of the algorithms that consider participating media is biased, simple averaging is not possible and the combination of results must be solved in a mathematically proven way.

Choosing the method which converges the fastest when rendering a scene is not straightforward. For example; the extent of the scene, the complexity of the objects, the type of medium or the lightsources have different effects on the speed and quality of each algorithm. The current research [Křivánek et al. 2014] tends to merge the strength of the currently available algorithms and to produce robust results for arbitrary scenes.

References

CEREZO, E., PÉREZ, F., PUEYO, X., SERON, F. J., AND SILLION, F. X. 2005. A survey on participating media rendering techniques. *The Visual Computer 21*, 5, 303–328.

CHANDRASEKHAR, S., 1960. Radiative transfer.

HACHISUKA, T., OGAKI, S., AND JENSEN, H. W. 2008. Progressive photon mapping. *ACM Transactions on Graphics (TOG) 27*, 5, 130.

HACHISUKA, T. 2013. Five common misconceptions about bias in light transport simulation.

JAKOB, W., REGG, C., AND JAROSZ, W. 2011. Progressive expectation-maximization for hierarchical volumetric photon mapping. In *Computer Graphics Forum*, vol. 30, Wiley Online Library, 1287–1297.

JAROSZ, W., DONNER, C., ZWICKER, M., AND JENSEN, H. W. 2008. Radiance caching for participating media. *ACM Transactions on Graphics (TOG) 27*, 1, 7.

JAROSZ, W., ZWICKER, M., AND JENSEN, H. W. 2008. The beam radiance estimate for volumetric photon mapping. In *ACM SIGGRAPH 2008 classes*, ACM, 3.

JAROSZ, W., NOWROUZEZAHRAI, D., SADEGHI, I., AND JENSEN, H. W. 2011. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics (TOG) 30*, 1, 5.

JAROSZ, W., NOWROUZEZAHRAI, D., THOMAS, R., SLOAN, P.-P., AND ZWICKER, M. 2011. Progressive photon beams. *ACM Transactions on Graphics (TOG) 30*, 6, 181.

JENSEN, H. W., AND CHRISTENSEN, P. H. 1998. Efficient simulation of light transport in scences with participating media using photon maps. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, 311–320.

KAJIYA, J. T. 1986. The rendering equation. In *ACM Siggraph Computer Graphics*, vol. 20, ACM, 143–150.

KELLER, A. 1997. Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 49–56.

KŘIVÁNEK, J., GEORGIEV, I., HACHISUKA, T., VÉVODA, P., ŠIK, M., NOWROUZEZAHRAI, D., AND JAROSZ, W. 2014. Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics (TOG) 33*, 4, 103.

NOVÁK, J., NOWROUZEZAHRAI, D., DACHSBACHER, C., AND JAROSZ, W. 2012. Progressive virtual beam lights. In *Computer Graphics Forum*, vol. 31, Wiley Online Library, 1407–1413.

NOVÁK, J., NOWROUZEZAHRAI, D., DACHSBACHER, C., AND JAROSZ, W. 2012. Virtual ray lights for rendering scenes with participating media. *ACM Transactions on Graphics (TOG) 31*, 4, 60.

PHONG, B. T. 1975. Illumination for computer generated pictures. *Communications of the ACM 18*, 6, 311–317.

SPENCER, B., AND JONES, M. W. 2009. Hierarchical photon mapping. *Visualization and Computer Graphics, IEEE Transactions on 15*, 1, 49–61.

SPENCER, B., AND JONES, M. W. 2009. Into the blue: Better caustics through photon relaxation. In *Computer Graphics Forum*, vol. 28, Wiley Online Library, 319–328.

TARINI, M., CIGNONI, P., AND MONTANI, C. 2006. Ambient occlusion and edge cueing for enhancing real time molecular visualization. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5, 1237–1244.