

Importance Sampling Polygonal Lights in Participating Media

Keven Villeneuve



Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

April 2019

Submitted to McGill University in partial fulfilment of the requirements for the degree of
Master of Engineering.

© 2019 Keven Villeneuve

Abstract

The photorealistic images that we are so used to in our everyday life results from the accurate simulation of light scattering in a scene as defined by the rendering equation. Due to its simplicity and flexibility, path tracing is currently the most popular technique used to solve this equation. Unfortunately, its correctness comes at the price of variance, manifesting itself as noise on the resulting image. Getting rid of this noise is a very time-consuming process as thousands of samples need to be computed for every pixel of the image. For this reason, path tracing is normally implemented using various variance reduction schemes such as importance sampling.

The noise is even more problematic in the case of participating media like clouds, smoke and fire since a distance also has to be sampled into the medium before sampling a ray direction, thus requiring even more samples. Rendering polygonal and mesh lights in participating media is particularly challenging due to the fact that these light sources do not emit light from a single point in space but along all points on their surface. In addition, the illumination coming from those lights is affected by many terms of different nature such as the geometry, the transmittance and the phase function.

In this thesis we propose to express any polygonal light as an infinite set of oriented point lights at the surface of the polygonal light. We derive an analytical solution to importance sample the geometry term of one of those oriented point light. We then develop a semi-analytical solution to importance sample the product of its geometry, transmittance and phase function terms. Finally, we use a simple tabulation scheme on a small set of oriented point lights to importance sample the geometry, transmittance and the phase function of a polygonal light. We show that our new sampling scheme converges faster than the state of the art for polygonal lights in participating media and is directly applicable to the renderers used today in the films and visual effects industry. We also briefly present ongoing work on importance sampling mesh lights using a hierarchical lights clustering strategy that takes advantage of our newly derived solution for polygonal lights.

Sommaire

Les images photo-réalistes de notre vie de tous les jours proviennent de la simulation adéquate de la diffusion de la lumière dans une scène définie par l'équation du rendu. Étant donné sa simplicité et flexibilité, le path tracing est la technique la plus populaire utilisée pour résoudre cette équation. Malheureusement, son exactitude vient au prix de la variance, ce manifestant par du bruit sur l'image résultante. La suppression de ce bruit est un procédé très coûteux en calculs étant donné que plusieurs milliers d'échantillons doivent être calculés pour chaque pixel de l'image. Pour cette raison, la technique du path tracing est normalement implémentée en utilisant une variété de méthodes de réduction de variance telle que l'échantillonnage préférentiel.

Le bruit est encore plus problématique dans le cas des médias participatifs comme les nuages, la fumée et les flammes, étant donné qu'une distance dans le média doit être échantillonnée avant d'échantillonner la direction d'un rayon. Le rendu de sources de lumière polygonales et de lumière définies par un maillage dans des médias participatifs est particulièrement difficile, puisque ces sources de lumière n'émettent pas seulement de la lumière à partir d'un seul point dans l'espace, mais au long de tous les points de leur surface. De plus, l'illumination provenant de ces sources de lumière est affectée par plusieurs termes de différentes natures telles que la géométrie, la transmittance et la fonction de phase.

Dans cette thèse, nous proposons d'exprimer toute source de lumière polygonale comme un ensemble infini de lumières ponctuelles orientées à sa surface. Nous dérivons une solution analytique afin d'appliquer l'échantillonnage préférentiel sur le terme de géométrie de l'une de ces lumières ponctuelles orientées. Par la suite, nous dérivons une solution semi-analytique afin d'appliquer l'échantillonnage préférentiel au produit des termes de géométrie, de transmittance et de fonction de phase de celle-ci. Finalement, nous utilisons une méthode de tabulation sur un petit ensemble de lumières ponctuelles orientées pour appliquer l'échantillonnage préférentiel aux termes d'une source de lumière polygonale. Nous démontrons que notre nouvelle méthode d'échantillonnage converge plus rapidement que l'état de l'art en rendu de lumières polygonales dans des médias participatifs et qu'elle est directement applicable aux systèmes de rendu utilisés aujourd'hui dans l'industrie du cinéma et des effets spéciaux. Nous présentons également nos travaux de recherche en cours sur le rendu de lumières définies par un maillage en utilisant une stratégie de regroupement hiérarchique de sources de lumière qui profite de notre solution pour les lumières polygonales.

Acknowledgments

Firstly, I want to thank my thesis supervisor Derek Nowrouzezahrai for transmitting me his passion for rendering and for introducing me to the hectic world of Computer graphics research. Thank you for demanding such high standards in anything you are involved with while still staying a very understanding person. Thank you for truly believing in me and pushing me to do things that I did not even know I was capable of. I will miss your contagious energy and your unreasonable fascination for computers with a certain fruit on it.

I also want to give special thanks to Iliyan Georgiev for its tremendous help and support during this challenging research project. Thank you for genuinely taking the time to explain the many things I was struggling with and for sharing your world-renowned expertise with me. I hope to see you at the next top secret Global Illuminati (GI) retreat. I also have to thank Martin De Lasa for guiding me and inspiring me to do this master's degree. Thank you for referring me to Derek, I honestly don't think I would be here if it wasn't for you.

Of course I couldn't have done it without all the amazing people at the Graphics & Imaging Lab (GIL), especially my everyday labmates Joey, Sayantan, Fan, Nicolas and Damien. People which I could count on when I was confused with unfamiliar ideas and techniques or to provide emotional support when times were tough. Thanks to all of you for making this whole experience much more interesting and satisfying. Special mention to Damien and Joey for sharing their excitement for beautiful mathematics, I am looking forward to your papers guys.

Finally, I could not have survived this crazy adventure without the continuous support of my mom Pierrette, my dad Camil and my sister Cynthia. Special thanks to my dad who at first showed incomprehension to the idea of me doing a master's degree. This thesis is the culmination of the two things that you've pushed me so hard to become good at: English and mathematics. I bet you didn't expect that this small action of yours would go so far. Sincerely, thank you for seeing through when I did not know any better.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Thesis Outline	3
2	Background	4
2.1	Light Transport Fundamentals	4
2.1.1	Light	4
2.1.2	Radiometry	5
2.1.3	Rendering Equation	9
2.1.4	Ray Tracing	12
2.1.5	Path Tracing	12
2.1.6	Monte Carlo	13
2.1.7	Variance Reduction	14
2.2	Light Transport in Participating Media	18
2.2.1	Participating Media	18
2.2.2	Radiative Transfer Equation (RTE)	19
2.2.3	Volume Rendering Equation (VRE)	20
2.2.4	Volumetric Path Tracing	24
3	Related Work	26
3.1	Equi-angular Sampling	26
3.1.1	Point Lights	27
3.1.2	Sphere Lights	29
3.1.3	Polygonal Lights	30

3.1.4	Transmittance MIS	31
3.2	Joint Importance Sampling (JIS)	31
3.3	Lightcuts	33
4	Importance Sampling Polygonal Lights	35
4.1	Polygonal Lights in Participating Media	35
4.2	Geometry term	38
4.2.1	Polygonal Light	38
4.2.2	Point-normal Light	41
4.3	Transmittance	48
4.3.1	Analytical Solution	48
4.3.2	Semi-Analytical Solution	49
4.4	Phase Function	52
4.4.1	Analytical Solution	52
4.4.2	Semi-Analytical Solution	54
4.5	Importance Sampling Polygonal Lights	56
4.6	Importance Sampling Mesh Lights	58
4.6.1	Lights Clustering	58
4.6.2	Volumetric Lightcuts	59
5	Results	63
5.1	Implementation	63
5.2	Technique Comparison	64
5.2.1	Point-normal Light	65
5.2.2	Rectangular Light	67
5.3	Convergence Analysis	69
5.4	Newton Solver Analysis	70
6	Conclusion	71
6.1	Limitations	72
6.2	Future Work	73
A	Appendices	75
A.1	Taylor expansion of the transmittance terms	75

A.2 Polynomial coefficients of the transmittance CDF	76
A.3 Taylor expansion of the HG phase function	77
A.4 Polynomial coefficients of the HG phase function CDF	77
References	78

List of Figures

1.1	Scene from the movie <i>Coco</i> rendered using a path tracer	1
1.2	Scene from the movie <i>Moana</i> featuring participating media	3
2.1	One of the many light path in a simple scene	5
2.2	Flux per surface area	6
2.3	Solid angle	7
2.4	Differential solid angle	7
2.5	Lambert's cosine law	8
2.6	Radiance	8
2.7	Rendering equation	9
2.8	Cornell Box	11
2.9	Importance sampling	15
2.10	Inversion method	16
2.11	Participating media	18
2.12	Light interactions in a medium	19
2.13	Examples of light interactions in participating media	19
2.14	Volume rendering equation (VRE)	20
2.15	Cosine of Henyey-Greenstein	22
2.16	Single scattering in the volume rendering equation	23
2.17	Volumetric path tracing	24
3.1	Equi-angular sampling	27
3.2	Point light source in an homogeneous medium (16 spp)	28
3.3	Sphere light source in an homogeneous an medium (16 spp)	29

3.4	Rectangular light source in an homogeneous medium (256 spp)	30
3.5	Light path of two bounces in an homogeneous scattering medium	31
3.6	Anisotropic phase function in the equi-angular frame	32
3.7	Binary tree used to cluster the triangle lights in Lightcuts	34
3.8	Shading point under different lightcuts	34
4.1	Single scattering VRE	36
4.2	Subtended solid angle of a triangular light	39
4.3	Approximation of a polygonal light using point-normal lights	40
4.4	Projection of an angle to a distance in the equi-angular frame	42
4.5	Clamping of the CDF	43
4.6	Angular frames	44
4.7	Frame for the normal	45
4.8	Taylor approximation of transmittance	51
4.9	Anisotropic phase function in equi-angular frame	53
4.10	Taylor approximation of the HG phase function	55
4.11	Subtended solid angle of a sub-polygon	57
4.12	Lightcuts data structure	60
4.13	Lightcuts	62
5.1	One sample comparison of our technique on the point-normal light scene. . .	65
5.2	Converged images of every technique on the point-normal light scene. . . .	65
5.3	One sample comparison of our technique on the point-normal light scene. . .	66
5.4	Converged images of every technique on the point-normal light scene. . . .	66
5.5	One sample comparison of our technique on the rectangular light scene. . . .	67
5.6	Converged images of every technique on the rectangular light scene.	67
5.7	One sample comparison of our technique on the rectangular light scene. . . .	68
5.8	Converged images of every technique on the rectangular light scene.	68
5.9	Convergence rate of our technique compared to equi-angular sampling. . . .	69
5.11	Average iterations per pixel of the Newton-Raphson solver.	70

CHAPTER 1

Introduction



Fig. 1.1: Scene from the movie *Coco* rendered using a path tracer. *Image source:* [Pix17]

Focusing on the purely visual aspect, Rendering is one of the major subfield in Computer graphics where the simulation of complex lighting phenomena are used to synthesize realistic images. The most notable examples are in movies and video games (Figure 1.1 and Figure 1.2), but it is also widely used in many other fields such as architecture, scientific computing and medical visualizations. To create one of those image we say we render a scene. A scene in the Computer graphics sense is composed of objects, those can have different kind of materials such as wood, glass or even shiny metals. A scene also has one or many light

sources. Objects can reflect, refract and scatter light coming from those light sources in a specific and well defined way. Rendering is thus the technology used to simulate the behavior of light in a scene and as a result produces a photorealistic image that is indistinguishable (in theory) to what it would look like in reality.

Those photorealistic images are normally rendered using a technique called *path tracing*. This technique computes the illumination of a scene by tracing light paths using a stochastic process. While this method is accurate it also has the unfortunate side-effect of generating noise on the resulting image. Completely eliminating the noise is a very time-consuming process as many thousands samples are normally needed, as such various techniques have been developed to limit the amount of noise generated by path tracing. Nevertheless path tracing has been recently adopted by the films and visual effects industry due to its simplicity and flexibility. The media & entertainment industry alone is about 1.8 trillion US\$ [sta]. The average budget for a blockbuster movie in 2018 is 100 million US\$. Rendering time increases by about 1.2x every year even with a 2.2x increase in computing power per year.

Film	Year	Render time (Million hours)	Renderfarm cores
Cars 2	2011	100[rena]	12 500
Monster University	2013	160[renb]	24 000
Big Hero 6	2014	190[renc]	55 000

At the scale of today's movies production, every millisecond count. The usual way to render a scene is to consider the case where the light is not attenuated while it travels through the air. Rendering of objects in participating media such as clouds, fog, fire and explosions (See figure 1.2) is vastly different than rendering with objects in vacuum or in the air. This is notoriously difficult because the light is attenuated by the medium. The current techniques to render those scenes are very computationally expensive. We can distinguish between two types of light sources: point lights (the ones without a surface) and area lights (the ones with a surface). A point light is a single point in space that emits the same amount of light in all directions. An area light is a light of a defined shape such as a rectangle, every point along its surface emits light in a direction normal to its shape. An area light can have any shape imaginable, even a bunny. If an area light has such a complex shape we say it is a *mesh light*. Efficient techniques already exist to render scenes in participating media using point lights, but so far it has been a very challenging thing to do for area lights.

One of the most used light sources in the industry are flat polygonal lights such as a rectangle lights, but they are severely limited because they are very expensive to render. In this thesis, I propose a faster way to render polygonal and mesh lights in participating media that is directly applicable to industrial strength renderers.



Fig. 1.2: Scene from the movie *Moana* featuring participating media such as clouds, smoke and lava. *Image source:* [Wal16]

1.2 Thesis Outline

In *Chapter 2* we introduce the background knowledge and mathematical framework necessary to render scenes with participating media. *Chapter 3* briefly summarizes the current state of the art for polygonal and mesh lights rendering in participating media. *Chapter 4* presents the main contribution of this thesis: a new algorithm to render polygonal and mesh lights in participating media more efficiently than what is currently possible. In *Chapter 5* we show results of our algorithm in a physically based renderer similar to the one used today in the film and visual effects industry. Finally, *Chapter 6* exposes the limitations of our method and proposes ideas for future work.

CHAPTER 2

Background

In this chapter we expose the background knowledge necessary to render realistic images of scenes in participating media. We describe the mathematical framework and techniques used to simulate light scattering in complex scenes. Finally we explain how these techniques can be improved to reduce the noise in the resulting images.

2.1 Light Transport Fundamentals

Before explaining how to render scenes in participating media it is important to understand the basics involved in rendering scenes with only surfaces. *Light transport* is the theory that studies how to compute the illumination in a scene by simulating light scattering. Various algorithms that varies in accuracy and efficiency are possible to achieve light transport.

2.1.1 Light

To simulate the lighting effects on a scene we need to first rigorously define what light is. From physics we know that light can be modeled as electromagnetic radiation resulting from Maxwell's equations[Max63] but for most lighting phenomena that we care about (that has a visual impact) we can forget the wavelike nature of light altogether and instead approximate light propagation as rays. In this model, a ray of light follows Fermat's principle which states that a light ray is the path between two points that takes the least amount of time. More simply, a light ray is a line perpendicular to the light's wavefront that travels in straight lines.

As such, light sources emit light rays carrying radiant energy that travel into the scene. When a ray hits an object, part of its energy is reflected, refracted and absorbed, depending on the properties of the object's material. Rays can bounce many times all around the scene before hitting a detector such as a camera or an eye. The rays from a light source to the eye form then what is commonly called a *light path* (Figure 2.1). When and only when those rays end up in our eyes we are able to *see* the scene in all its details.

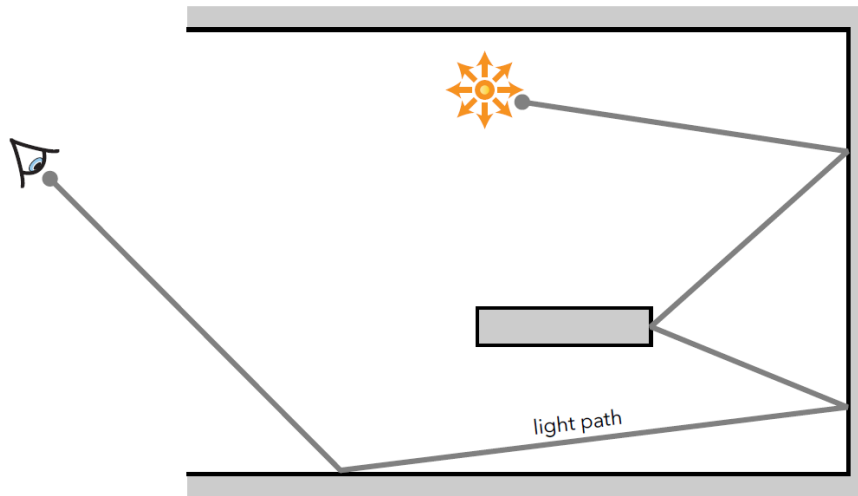


Fig. 2.1: One of the many light path from the light source to the camera in a simple scene.
Image source: [Now18]

2.1.2 Radiometry

To continue our description of light, we need to define the units used to measure it. *Radiometry* is the technique that measures electromagnetic radiation such as light as a distribution of the radiation's power in space. Contrary to *photometry*, it stays independent of the interaction with the human eye.

Flux

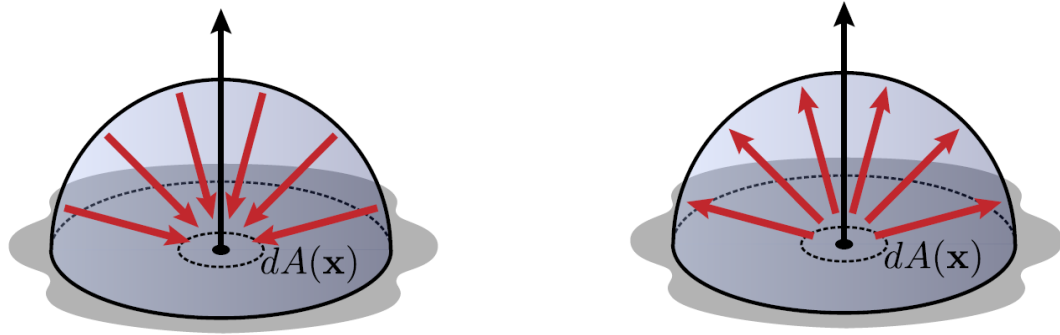
The total amount of radiant energy passing through a surface or space over time is defined as the *flux* $\Phi(A)$. This measure includes the radiant energy arriving but also leaving a surface and depends on the area of the surface. To make flux independent of the surface's area we use *Irradiance* or *Radiosity* instead. The flux in an infinitesimal patch $dA(\mathbf{x})$ **arriving** at

position $\mathbf{x}$ on a surface like seen on figure 2.2a is called *irradiance* $E(\mathbf{x})$:

$$E(\mathbf{x}) = \frac{d\Phi(A)}{dA(\mathbf{x})} \left[\frac{W}{m^2} \right], \quad (2.1)$$

while the flux in an infinitesimal patch **leaving** a surface (See figure 2.2b) is called *radiosity* $B(\mathbf{x})$:

$$B(\mathbf{x}) = \frac{d\Phi(A)}{dA(\mathbf{x})} \left[\frac{W}{m^2} \right]. \quad (2.2)$$



(a) Irradiance is the flux arriving at a surface. (b) Radiosity is the flux leaving a surface.

Fig. 2.2: Flux per surface area. *Image source:* [Now18]

Radiance

Both Irradiance and Radiosity measures the flux per area arriving/leaving a surface regardless of the direction from which they are coming from. In rendering we want a measure that is also independent of the direction (ω), as such we use *Radiance* $L(\mathbf{x}, \omega)$ as the main quantity for light transport. To define such measure we first need to define the flux in an infinitesimal cone of directions. This cone is the equivalent of an angle in 3D and is called a *solid angle* Ω (See figure 2.3). It is the ratio between the area of a surface A on a sphere of radius r :

$$\Omega = \frac{A}{r^2}. \quad (2.3)$$

The solid angle unit of measure is called the steradian (sr). If the surface A covers the whole sphere then $\Omega = 4\pi sr$.

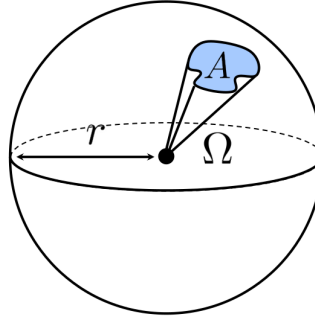


Fig. 2.3: Solid angle. *Image source:* [Now18]

To compute the flux per direction we need to define the differential solid angle $d\Omega$:

$$d\Omega = \frac{dA}{r^2}. \quad (2.4)$$

On figure 2.4 we can see that $dA = (rd\theta)(r \sin \theta d\phi)$. By using this equation we can express the differential solid angle as:

$$d\Omega = \frac{(rd\theta)(r \sin \theta d\phi)}{r^2}. \quad (2.5)$$

If we now only consider the case of a unit sphere ($r = 1$), then $d\Omega = \sin \theta d\theta d\phi = d\omega$ and the solid angle can be equivalently defined as:

$$\Omega = \int_{S^2} d\omega. \quad (2.6)$$

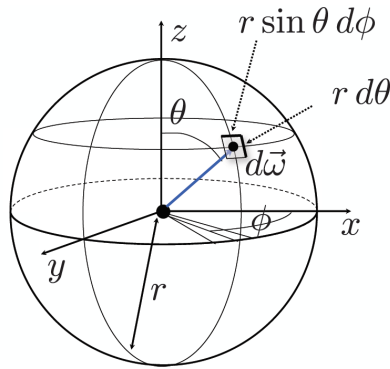


Fig. 2.4: Differential solid angle. *Image source:* [Now18]

The flux per solid angle per perpendicular area $dA^\perp(\mathbf{x}, \boldsymbol{\omega})$ is called *radiance*:

$$L(\mathbf{x}, \boldsymbol{\omega}) = \frac{d^2\Phi(A)}{d\boldsymbol{\omega}dA^\perp(\mathbf{x}, \boldsymbol{\omega})} = \frac{d^2\Phi(A)}{d\boldsymbol{\omega}dA(\mathbf{x}) \cos \theta} \left[\frac{W}{m^2 sr} \right]. \quad (2.7)$$

It is defined per perpendicular area instead of simply per area since electromagnetic radiation like light follows Lambert's cosine law which states that only the radiation that is directly perpendicular to a surface will contribute. If the radiation arrives not exactly perpendicularly then it will be modulated by the cosine of the angle θ between the surface and the radiation's direction like seen on figure 2.5 and 2.6.

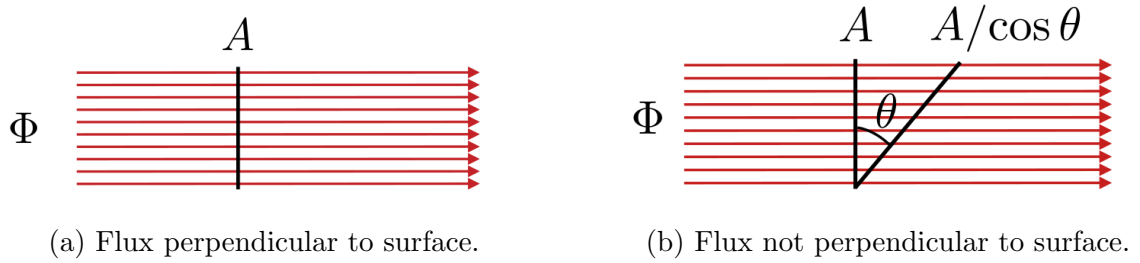


Fig. 2.5: Lambert's cosine law. *Image source:* [Now18]

Radiance is the fundamental quantity used in physically based rendering. In reality radiance is defined per wavelength but for rendering purposes it is often sufficient to approximate it as three values: red (R), green (G) and blue (B).

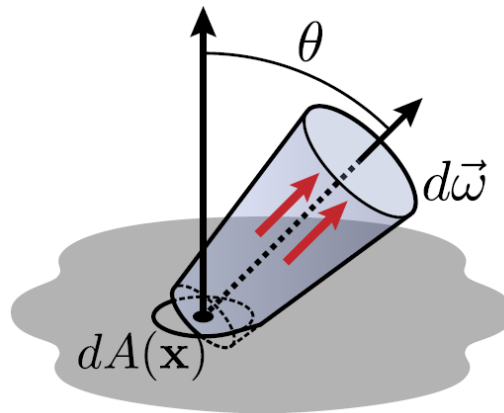


Fig. 2.6: Radiance at a point $\mathbf{x}$ in a direction $\boldsymbol{\omega}$. *Image source:* [Now18]

2.1.3 Rendering Equation

By applying the ray optics approximation and the law of conservation of energy, the outgoing radiance can be expressed as the sum of emitted radiance and reflected radiance:

$$L(\mathbf{x}, \boldsymbol{\omega}) = L_e(\mathbf{x}, \boldsymbol{\omega}) + L_r(\mathbf{x}, \boldsymbol{\omega}). \quad (2.8)$$

Assuming that light travels in vacuum, as to be not attenuated by the medium and as such radiance remains constant along a ray, leads to the famous *Rendering Equation* first derived by Kajiya et al. in 1986 [Kaj86]:

$$L(\mathbf{x}, \boldsymbol{\omega}) = L_e(\mathbf{x}, \boldsymbol{\omega}) + \int_{S^2} f_r(\mathbf{x}, \boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) L_i(\mathbf{x}, \bar{\boldsymbol{\omega}}) |n(\mathbf{x}) \cdot \bar{\boldsymbol{\omega}}| d\bar{\boldsymbol{\omega}}. \quad (2.9)$$

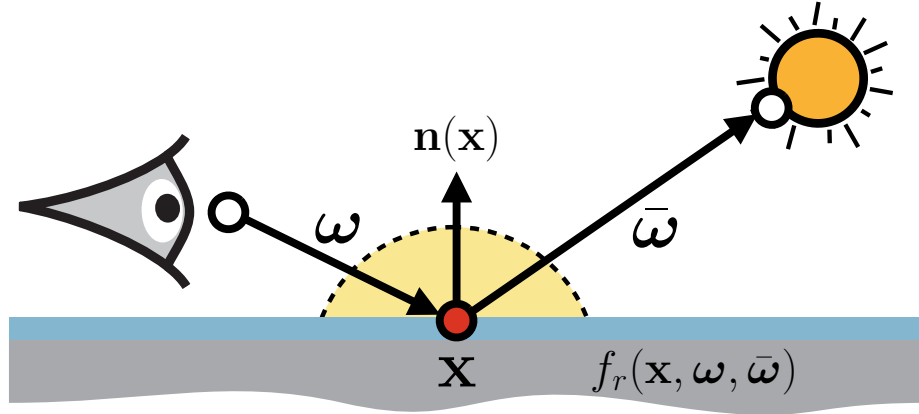


Fig. 2.7: Rendering equation.

By solving this equation we can generate a physically correct image of a scene seen from the point of view of a camera. It says exactly what color should a given pixel be. The integral from this equation implies that the radiance coming from every direction around the shading point $\mathbf{x}$ needs to be taken into account to compute the reflected radiance. This integral is composed of three parts: the BSDF $f_r(\mathbf{x}, \boldsymbol{\omega}, \bar{\boldsymbol{\omega}})$, the incoming light $L_i(\mathbf{x}, \bar{\boldsymbol{\omega}})$ and the cosine term $|n(\mathbf{x}) \cdot \bar{\boldsymbol{\omega}}|$. We use the absolute value symbol to represent the clamp function whenever we are talking about dot products like the one in the rendering equation.

BSDF

The BSDF (Bidirectional Scattering Distribution Function) is a mathematical model of surface reflection for various materials like metals, plastics, glass, etc. It defines how it affects light. When an incoming ray $\bar{\omega}$ hits a material, part of its radiance is reflected and part of it is refracted according to the Fresnel equations (2.10):

$$\begin{aligned}
 p_{\parallel} &= \frac{\eta_2 \cos \theta_1 - \eta_1 \cos \theta_2}{\eta_2 \cos \theta_1 + \eta_1 \cos \theta_2} \\
 p_{\perp} &= \frac{\eta_1 \cos \theta_1 - \eta_2 \cos \theta_2}{\eta_1 \cos \theta_1 + \eta_2 \cos \theta_2} \\
 \text{Reflected: } F_r &= \frac{1}{2} \left(p_{\parallel}^2 + p_{\perp}^2 \right) \\
 \text{Refracted: } F_t &= 1 - F_r
 \end{aligned} \tag{2.10}$$

To satisfy physically-correct properties a BSDF needs to respect two principles:

$$f_r(\mathbf{x}, \boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) = f_r(\mathbf{x}, \bar{\boldsymbol{\omega}}, \boldsymbol{\omega}) \quad \text{Helmholtz reciprocity} \tag{2.11}$$

$$\int_{H^2} f_r(\mathbf{x}, \boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) |n(\mathbf{x}) \cdot \bar{\boldsymbol{\omega}}| d\bar{\boldsymbol{\omega}} \leq 1, \quad \forall \boldsymbol{\omega} \quad \text{Energy conservation} \tag{2.12}$$

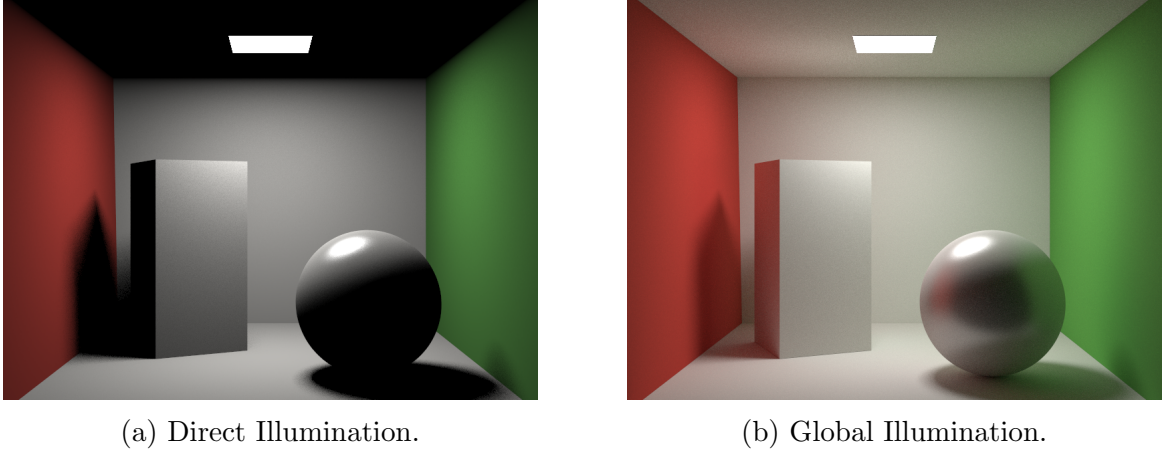
Many different models exist but the most popular are the Microfacet models. Since this research project focus exclusively on light transport, we will not go into more details even though their study is no less important.

Incoming light

The light can be coming directly from a light source, in this case it is called *direct illumination* (DI):

$$L_i(\mathbf{x}, \bar{\boldsymbol{\omega}}) = L_e(r(x, \bar{\boldsymbol{\omega}}), -\bar{\boldsymbol{\omega}}). \tag{2.13}$$

The $r(x, \bar{\boldsymbol{\omega}})$ term is the ray tracing operation that resolves into the position where the given ray hits a surface in the scene. In equation 2.13 it resolves into a position on the light source. If the light is coming indirectly from the light bouncing on objects in the scene it is known as *global illumination* (GI). In this case, the rendering equation becomes recursive. As we can see in figure 2.8, a lot of the realism that we are so used to in our daily life comes from the accurate computation of global illumination.

**Fig. 2.8:** Cornell Box.

Cosine term

This last part is necessary to make the physics work and stay consistent with our definition of radiance. If the incoming ray arrives with a direction $\bar{\omega}$ not exactly perpendicular to the surface, only a fraction of the radiance will be reflected. This fraction is the cosine of the angle between the incoming ray direction and the surface's normal $n(\mathbf{x})$ at the shading point $\mathbf{x}$ or equivalently the dot product between them. We also need to make sure that the cosine always stays positive by clamping its value to zero. Not doing so would results in negative radiance which is not physically possible.

Area formulation

The rendering equation can also be expressed as an integral over surface area instead of over directions. We thus do a change of variable using the Jacobian

$$d\bar{\omega} = \frac{|n(\mathbf{y}) \cdot -\bar{\omega}|}{\|\mathbf{x} - \mathbf{y}\|^2} dA(\mathbf{y}), \quad (2.14)$$

and obtain the following formulation:

$$\begin{aligned} L(\mathbf{x}, \mathbf{z}) &= L_e(\mathbf{x}, \mathbf{z}) + \int_{A_e} f_r(\mathbf{x}, \mathbf{y}, \mathbf{z}) L_i(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) dA(\mathbf{y}) \\ G(\mathbf{x}, \mathbf{y}) &= V(\mathbf{x}, \mathbf{y}) \frac{|n(\mathbf{x}) \cdot \bar{\omega}| |n(\mathbf{y}) \cdot -\bar{\omega}|}{\|\mathbf{x} - \mathbf{y}\|^2}. \end{aligned} \quad (2.15)$$

The $V(\mathbf{x}, \mathbf{y})$ term is the equivalent of the ray tracing operation, it has a value of one if there is nothing in between point $\mathbf{x}$ and $\mathbf{y}$ and as such the two points are mutually visible. If they are not visible then the function returns zero.

2.1.4 Ray Tracing

In order to render a scene we need to compute the value of each pixel of the image of the scene seen from the point of view of a camera using the rendering equation. For each pixel of the image we trace a ray from the camera to the scene in the camera direction $\boldsymbol{\omega}$. We then compute the radiance at the hit point $\mathbf{x}$ of this ray with the scene using the rendering equation which results in the corresponding color of the pixel. We can repeat this procedure many times for each pixel and average their value to reduce aliasing and variance.

2.1.5 Path Tracing

Path tracing is a specific flavor of ray tracing that uses Monte Carlo to trace light paths. To compute the correct color of a pixel, all the light paths between the light sources and the camera need to be taken into account. Since it would be unfeasible to trace every possible light paths, we use a stochastic process to generate only a few of them and average their contribution. This gives as a result an estimate of the correct radiance:

$$L(\mathbf{x}, \boldsymbol{\omega}) \approx L_e(\mathbf{x}, \boldsymbol{\omega}) + \frac{1}{N} \sum_{i=1}^N \frac{f_r(\mathbf{x}, \boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) L_i(\mathbf{x}, \bar{\boldsymbol{\omega}}) |n(\mathbf{x}) \cdot \bar{\boldsymbol{\omega}}|}{p(\bar{\boldsymbol{\omega}})}. \quad (2.16)$$

This method is unbiased, meaning it converges to the right answer as $N \rightarrow \infty$. At any point the resulting image has bias, only variance manifesting itself as noise. This noise is an unfortunate side effect of Monte Carlo methods, but recently the movie industry completely switched to this technique due to its simplicity and flexibility.

2.1.6 Monte Carlo

Monte Carlo is a numerical integration technique that uses random numbers to compute an integral. As such, it is important to review some notions on probabilities.

Random variable

A random variable X is a variable that randomly takes a value according to a given probability distribution. These distributions are defined by their CDF and PDF.

Probability distribution

The PDF or *Probability density function* is a function that defines the probability of a random variable to take a given value:

$$p(x) = \frac{dP(x)}{dx} \quad p(x) \geq 0 \quad \int p(x) = 1, \quad (2.17)$$

where $P(x)$ is the CDF or *Cumulative distribution function*. The CDF is defined as:

$$P(x) = \text{Prob}\{X \leq x\} = \int_0^x p(x') dx' \quad P(x) \in [0, 1]. \quad (2.18)$$

The probability for a random variable X to be between a value a and b can be computed using:

$$\text{Prob}\{a \leq X \leq b\} = \int_a^b p(x) dx = P(b) - P(a). \quad (2.19)$$

The canonical probability distribution is called the *uniform distribution*, its PDF is defined as:

$$p(x) = \begin{cases} 1 & x \in [0, 1], \\ 0 & \text{otherwise.} \end{cases} \quad (2.20)$$

A random variable from this distribution is commonly referred to as ξ .

Expected value and variance

Two important statistics on a probability distribution can be calculated, its expected value $E[Y]$ and its variance $V[Y]$. Given a function $Y = f(X)$, its expected value is:

$$E[Y] = \int_D f(x)p(x) dx, \quad (2.21)$$

and its variance is:

$$V[Y] = E[(Y - E[Y])^2] = E[Y^2] - E[Y]^2. \quad (2.22)$$

Monte Carlo estimator

Given an integral:

$$F = \int_a^b f(x) dx, \quad (2.23)$$

one can define its Monte Carlo estimator:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}. \quad (2.24)$$

This estimator is unbiased since:

$$E[F_N] = \int_a^b f(x) dx. \quad (2.25)$$

It follows that to compute a Monte Carlo estimator, we need to generate samples x_i according to a PDF $p(x)$ and average their contribution. The simplest PDF to use is the uniform one:

$$p(x) = \frac{1}{b - a}. \quad (2.26)$$

2.1.7 Variance Reduction

While Monte Carlo integration allows us to solve the rendering equation in a straightforward and elegant manner without introducing any bias, it converges quite slowly. Reducing the variance (noise) by a factor of two requires four times as many samples. As such, when using Monte Carlo one needs to use an efficient variance reduction strategy. There are various ways that this can be accomplished, one very popular option is to use *importance sampling*.

Importance sampling

Very often, only a small region of the domain will actually contribute to a Monte Carlo estimator. Importance sampling allows to sample more efficiently by making it more likely to sample into the region where the contribution is the highest. In the particular case where the PDF of the Monte Carlo estimator of equation 2.24 is defined to be:

$$\begin{aligned} p(x) &= cf(x) \\ c &= \frac{1}{\int_a^b f(x) dx}. \end{aligned} \quad (2.27)$$

The resulting Monte Carlo estimator has zero variance or no noise, meaning that the correct value can be obtained using only one sample:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)} = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{cf(X_i)} = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{\frac{1}{\int_a^b f(x) dx} f(X_i)} = \int_a^b f(x) dx. \quad (2.28)$$

The right part $f(x)$ cancels the noise for any sample and the left part c returns the correct value of the integral. The constant c is commonly known as the *normalization factor*. This is essentially the most efficient way to sample this function. While this is promising, it is normally very difficult to obtain as to derive this perfect importance sampling scheme, the original integral needs to be solved analytically. As such, it is common to sample according to only one part of the integrand. If the PDF is similar to the integrand, the variance of the estimator is significantly reduced (See figure 2.9). When designing a new importance sampling scheme, any function can be used as a PDF but the PDF should never be zero where the integrand is not zero and it should never be negative, otherwise the estimator will be biased.

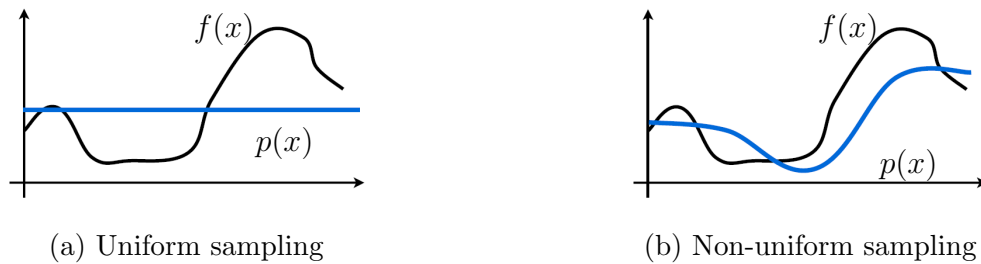


Fig. 2.9: Importance sampling. *Image source:* [Now18]

Inversion method

The *inversion method* is the usual procedure to derive a new importance sampling scheme. The first step is to compute the CDF of the desired PDF and normalize it:

$$P(x) = c \int_0^x p(x') dx' \quad (2.29)$$

$$c = \frac{1}{\int_{x_{\min}}^{x_{\max}} p(x') dx'}. \quad (2.30)$$

Then the CDF is inverted to sample according to the PDF (See figure 2.10):

$$X_i = P^{-1}(\xi). \quad (2.31)$$

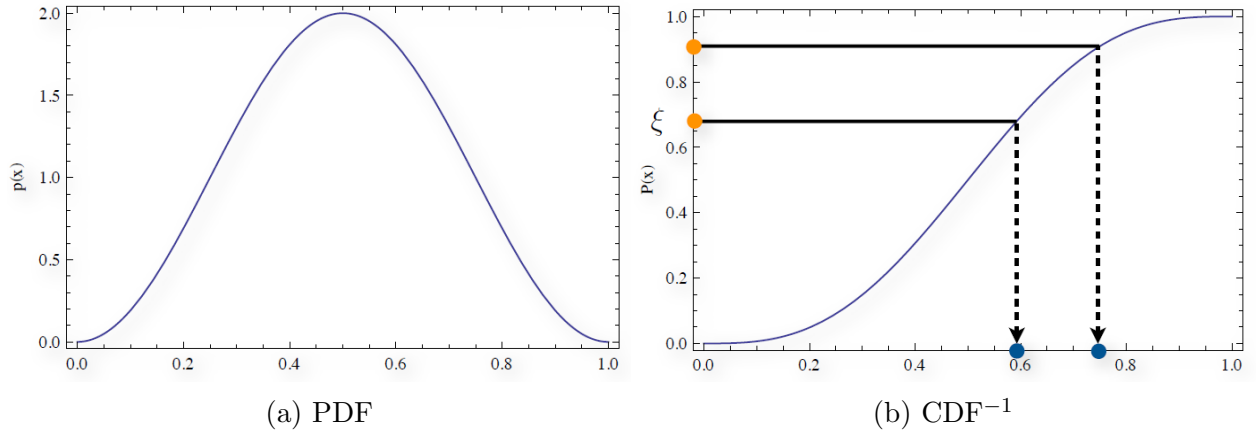


Fig. 2.10: Inversion method. *Image source:* [Now18]

Transforming between PDFs

When designing importance sampling schemes it is often necessary to transform from one distribution to another. Given T a one-to-one (bijective) transformation, the PDF of a distribution $Y_i = T(X_i)$ is

$$p_y(y) = p_y(T(x)) = \frac{p_x(x)}{|J_T(x)|}, \quad (2.32)$$

where $|J_T(x)|$ is the absolute value of the determinant of the Jacobian of T .

Direction sampling

To solve the rendering equation (2.9) using a Monte Carlo estimator, a random direction in a sphere around the shading point needs to be sampled. There are various ways to accomplish this. One can importance sample according to the BSDF. A different importance sampling scheme have been derived for almost every BSDF, often those also importance sample the cosine term. Another way is to importance sample according to the direct incoming light. The general way to do this is to sample uniformly the area of the light source. This has the advantage of working with light sources of any shape. A better way is to importance sample according to the subtended solid angle of the light source. Using this strategy, no sample is wasted since only the region that is guaranteed to contribute is sampled. While this is optimal, the sampling scheme changes for every light shape. Most of the simplest shapes (sphere, rectangle, triangle...) have already been derived.

Product sampling and MIS

While we have described many ways to importance sample the individual parts of the rendering equation, the best way would be to importance sample all the parts at once (i.e. their product), but in practice it is often very difficult to do or too slow to be usable. The next best thing is to sample from the average PDF of each part. This is what Veach introduced with *Multiple Importance Sampling (MIS)* in 1997 [Vea97]. A Monte Carlo estimator can be splitted into two estimators, each targetting a different part of the integrand $f(x)$:

$$F^{N_1+N_2} = \frac{1}{N_1} \sum_{i=1}^{N_1} w_1(x_i) \frac{f(x_i)}{p_1(x_i)} + \frac{1}{N_2} \sum_{j=1}^{N_2} w_2(x_j) \frac{f(x_j)}{p_2(x_j)}, \quad (2.33)$$

where $w_1(x) + w_2(x) = 1$. Each estimator can be sampled according to a different strategy corresponding to different PDF $p_1(x)$ and $p_2(x)$. The balance heuristic:

$$w_s(x) = \frac{N_s p_s(x)}{N_1 p_1(x) + N_2 p_2(x)}, \quad (2.34)$$

can be used to sample from the average PDF. Shown here to sample from the average of two strategies, but this sampling scheme can be trivially generalized to M strategies.

2.2 Light Transport in Participating Media

The light transport theory that we have developed in the previous sections is only applicable to scenes composed of hard surfaces. To handle scenes where the light can also be attenuated by various participating media we need to extend it.

2.2.1 Participating Media

Light travelling in vacuum is not attenuated while it goes through space. Similarly, the light that travels through the air is normally not very attenuated either, as such a common assumption in physically based rendering is to consider the air as a vacuum. When light encounters other media such as fog, clouds and smoke (See figure 2.11) then this approximation breaks down and the rendering equation used previously does not produce accurate images anymore. Media that affect light propagation in such way are commonly called *participating media*. In the scenes of figure 2.11, light is affected by the vast number of microscopic



(a) Fog. *Image source:* [Ste] (b) Clouds. *Image source:* [med] (c) Smoke & fire. *Image:* [wik]

Fig. 2.11: Participating media.

particles forming the medium in which light travels. Since it would be unfeasible to explicitly model every particle of a medium, it is defined statistically via a few coefficients μ and densities. It follows that more light will be able to pass through a medium with low density than a very dense one for which most of the light will be obstructed by the many microscopic particles of the medium. Participating media are generally divided into two types: *homogeneous* and *heterogeneous*. The coefficients of the homogeneous ones are the same everywhere, while they vary in space for heterogeneous media. Many of the natural phenomena such as fog, clouds and smoke are heterogeneous media but we can think of a few real world examples like a glass of orange juice as being nearly homogeneous.

2.2.2 Radiative Transfer Equation (RTE)

When light travels into a medium, radiance changes along the ray. As such, the rendering equation needs to be rederived. There are four types of light interactions within a medium: absorption, out-scattering, in-scattering and emission (See figure 2.12).

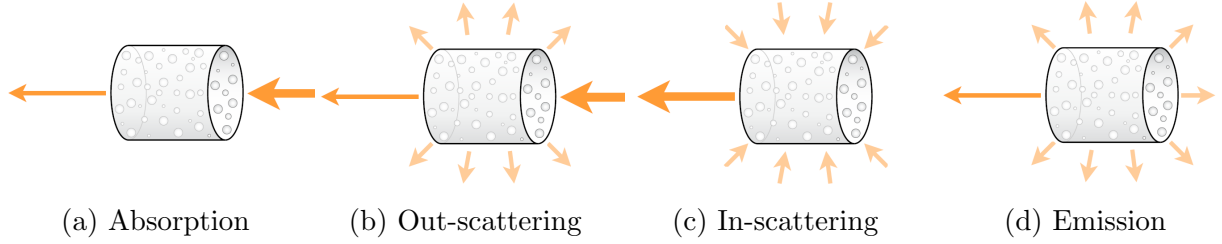


Fig. 2.12: Light interactions in a medium. *Image source:* [Now18]



(a) Smoke (absorption). (b) Clouds (scattering). (c) Explosion (emission).

Fig. 2.13: Examples of light interactions in participating media. *Image source:* [wik]

On figure 2.13a we can see how most of the light is absorbed by the smoke, on figure 2.13b we see scattering of light that gives clouds its characteristic appearance and finally figure 2.13c shows an explosion, a great example of light emission in participating media. The change in radiance traveling in a direction ω through a differential beam segment of the medium at point $\mathbf{x}$ can be described by the *radiative transfer equation (RTE)* [Cha13]:

$$(\omega \cdot \nabla)L(\mathbf{x}, \omega) = -\mu_a(\mathbf{x})L(\mathbf{x}, \omega) - \mu_s(\mathbf{x})L(\mathbf{x}, \omega) + \mu_s(\mathbf{x})L_s(\mathbf{x}, \omega) + \mu_a(\mathbf{x})L_e(\mathbf{x}, \omega). \quad (2.35)$$

The $L_s(\mathbf{x}, \omega)$ term is called the *in-scattered radiance* and will be formally defined later. This term takes into account the light coming from all directions at a point $\mathbf{x}$ in a medium.

2.2.3 Volume Rendering Equation (VRE)

Integrating both sides of the differential RTE leads to:

$$L(\mathbf{x}, \boldsymbol{\omega}) = \int_0^\infty T(\mathbf{x}, \mathbf{y}) [\mu_a(\mathbf{y}) L_e(\mathbf{y}, \boldsymbol{\omega}) + \mu_s(\mathbf{y}) L_s(\mathbf{y}, \boldsymbol{\omega})] d\mathbf{y}. \quad (2.36)$$

We denote distances in *italic* and points in **bold**. To take into account scenes that also contain objects with surfaces, this equation needs to be extended. The rendering equation 2.39 bounds equation 2.36 to the nearest surface at distance z , leading to the *volume rendering equation (VRE)*:

$$L(\mathbf{x}, \boldsymbol{\omega}) = \int_0^z T(\mathbf{x}, \mathbf{y}) [\mu_a(\mathbf{y}) L_e(\mathbf{y}, \boldsymbol{\omega}) + \mu_s(\mathbf{y}) L_s(\mathbf{y}, \boldsymbol{\omega})] d\mathbf{y} + T(\mathbf{x}, \mathbf{z}) L(\mathbf{z}, \boldsymbol{\omega}) \quad (2.37)$$

$$L_s(\mathbf{y}, \boldsymbol{\omega}) = \int_{S^2} f_p(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) L_i(\mathbf{y}, \bar{\boldsymbol{\omega}}) d\bar{\boldsymbol{\omega}} \quad (2.38)$$

$$L(\mathbf{z}, \boldsymbol{\omega}) = L_e(\mathbf{z}, \boldsymbol{\omega}) + \int_{S^2} f_r(\mathbf{z}, \boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) L_i(\mathbf{z}, \bar{\boldsymbol{\omega}}) |n(\mathbf{z}) \cdot \bar{\boldsymbol{\omega}}| d\bar{\boldsymbol{\omega}}. \quad (2.39)$$

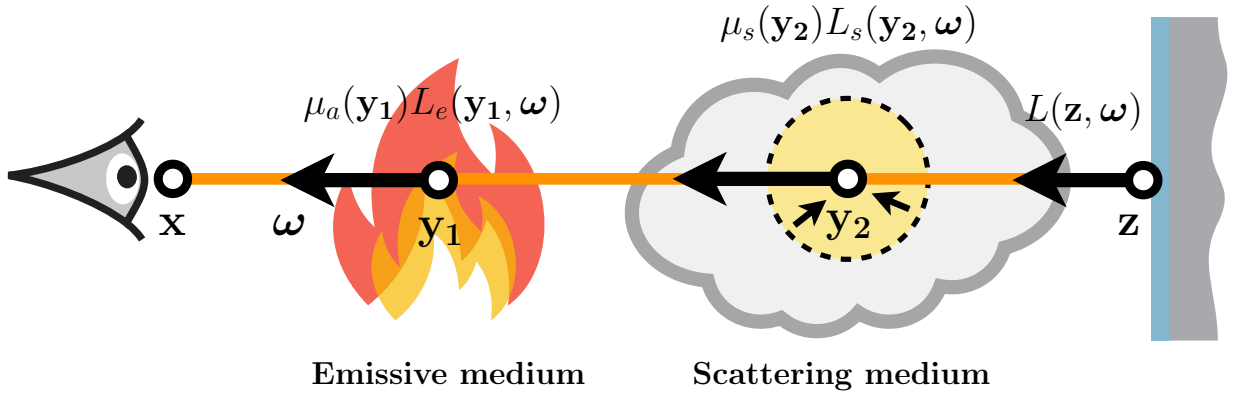


Fig. 2.14: Volume rendering equation (VRE).

The nested integral makes the VRE much more challenging to solve than the rendering equation, since light can come from any direction at any depth into the medium. The VRE involves many terms of different nature: the transmittance $T(\mathbf{x}, \mathbf{y})$, the phase function $f_p(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}})$ and the incoming light $L_i(\mathbf{y}, \bar{\boldsymbol{\omega}})$.

Transmittance

The transmittance is the term that specifies how the light is attenuated while it travels through the medium. It can be seen as the fractional visibility between two points and is essentially an exponential decay. If we take all the radiance losses from the RTE:

$$(\boldsymbol{\omega} \cdot \nabla)L(\mathbf{x}, \boldsymbol{\omega}) = -\mu_a(\mathbf{x})L(\mathbf{x}, \boldsymbol{\omega}) - \mu_s(\mathbf{x})L(\mathbf{x}, \boldsymbol{\omega}) \quad (2.40)$$

$$= -(\mu_a(\mathbf{y}) + \mu_s(\mathbf{y}))L(\mathbf{x}, \boldsymbol{\omega}) \quad (2.41)$$

$$= -\mu_t(\mathbf{x})L(\mathbf{x}, \boldsymbol{\omega}). \quad (2.42)$$

For the sake of demonstration we assume an homogeneous medium which has a constant $\mu_t(\mathbf{x})$ along a finite beam of the medium:

$$\frac{dL(\mathbf{x}, \boldsymbol{\omega})}{L(\mathbf{x}, \boldsymbol{\omega})} = -\mu_t(\mathbf{x})dy. \quad (2.43)$$

By integrating equation 2.43 along the beam from 0 to y we obtain:

$$\frac{L_y}{L_0} = e^{-\mu_t y}. \quad (2.44)$$

This equation is known as the *Beer-Lambert law*[Lam60]. In the context of the VRE we call this ratio transmittance:

$$T(y) = e^{-\mu_t y} \quad (2.45)$$

$$T(\mathbf{x}, \mathbf{y}) = e^{-\mu_t \|\mathbf{x} - \mathbf{y}\|}. \quad (2.46)$$

A similar derivation can be done to obtain the equivalent transmittance expression for heterogeneous media for which their μ_s , μ_a and μ_t coefficients can vary in space:

$$T(y) = e^{-\tau(y)} = e^{-\int_0^y \mu_t(\mathbf{x} - s\boldsymbol{\omega}) ds} \quad (2.47)$$

$$T(\mathbf{x}, \mathbf{y}) = e^{-\int_0^y \mu_t(\mathbf{x} - s\boldsymbol{\omega}) ds}. \quad (2.48)$$

The term $\tau(y)$ is called the *optical thickness*. The integral involved in the optical thickness makes heterogeneous media much more challenging than homogeneous ones.

Phase function

The phase function is the analog of the BSDF in participating media. It is a probability distribution that describes how the medium scatters light. To be physically correct a phase function must obey the principle of conservation of energy:

$$\int_{S^2} f_p(\omega, \bar{\omega}) d\bar{\omega} = 1. \quad (2.49)$$

We distinguish between two types: *isotropic* and *anisotropic*.

Isotropic

A medium with an isotropic phase function scatters light uniformly in all directions.

$$f_p(\omega, \bar{\omega}) = \frac{1}{4\pi} \quad (2.50)$$

Anisotropic

A medium with an anisotropic phase function favors scattering light in a particular direction. The most popular is the *Heney-Greenstein (HG)* phase function[HG41]:

$$f_p(\theta) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g \cos \theta)^{3/2}} \quad (2.51)$$

$g = 0$: *isotropic scattering*

$g > 0$: *forward scattering*

$g < 0$: *backward scattering,*

where $g \in [-1, 1]$ is the *asymmetry coefficient* and $\cos \theta = -\omega \cdot \bar{\omega}$ (See figure 2.15).

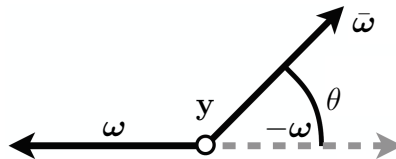


Fig. 2.15: Cosine of Heney-Greenstein. *Image source:* [Now18]

Incoming light

Similarly to the rendering equation, if the incoming light $L_i(\mathbf{y}, \bar{\omega})$ arrives directly from a light source (See figure 2.16), then it is known as *single scattering*:

$$L_i(\mathbf{y}, \bar{\omega}) = T(\mathbf{y}, r(\mathbf{y}, \bar{\omega}))L_e(r(\mathbf{y}, \bar{\omega}), -\bar{\omega}) = T(\mathbf{y}, \mathbf{u})L_e(\mathbf{u}, -\bar{\omega}). \quad (2.52)$$

If the light is coming indirectly from light that scattered multiple times inside the medium, it is known as *multiple scattering* and the volume rendering equation becomes recursive.

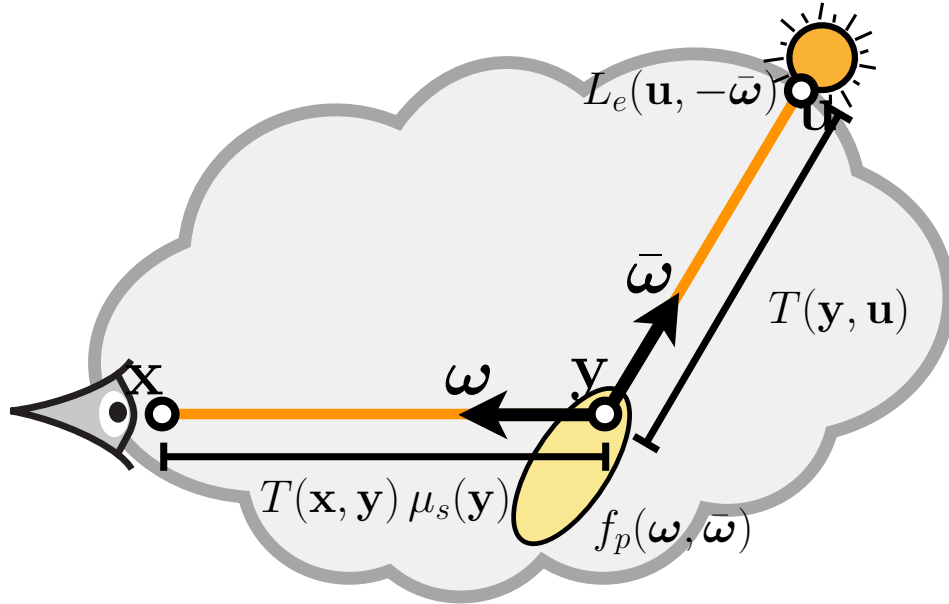


Fig. 2.16: Single scattering in the volume rendering equation.

Area formulation

Similarly to the rendering equation, we can derive an area formulation of the VRE:

$$L(\mathbf{x}, \mathbf{z}) = \int_0^z T(\mathbf{x}, \mathbf{y}) [\mu_a(\mathbf{y})L_e(\mathbf{y}, \mathbf{x}) + \mu_s(\mathbf{y})L_s(\mathbf{y}, \mathbf{x})] d\mathbf{y} + T(\mathbf{x}, \mathbf{z})L(\mathbf{z}, \mathbf{x}) \quad (2.53)$$

$$L_s(\mathbf{y}, \mathbf{x}) = \int_{A_e} f_p(\mathbf{y}, \mathbf{x}, \mathbf{u})L_i(\mathbf{y}, \mathbf{u})V(\mathbf{y}, \mathbf{u})G(\mathbf{y}, \mathbf{u}) dA(\mathbf{u}) \quad (2.54)$$

$$G(\mathbf{y}, \mathbf{u}) = \frac{|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|}{\|\mathbf{y} - \mathbf{u}\|^2} \quad (2.55)$$

2.2.4 Volumetric Path Tracing

The Monte Carlo estimator of the VRE (without surfaces) is:

$$L(\mathbf{x}, \omega) \approx \frac{1}{N} \sum_{i=1}^N \frac{T(\mathbf{x}, \mathbf{y}) [\mu_a(\mathbf{y}) L_e(\mathbf{y}, \omega) + \mu_s(\mathbf{y}) f_p(\omega, \bar{\omega}) L_i(\mathbf{y}, \bar{\omega})]}{p(y) p(\bar{\omega})}. \quad (2.56)$$

To solve the VRE using path tracing we need to first sample a distance y into the medium and then sample a direction $\bar{\omega}$. This process is continued until the light path ends on the light source or exit the scene. On figure 2.17 we see a light path of 4 bounces, the light ray enters a medium, scatters inside it, bounce on a surface and re-enters the medium to finally hit the light source.

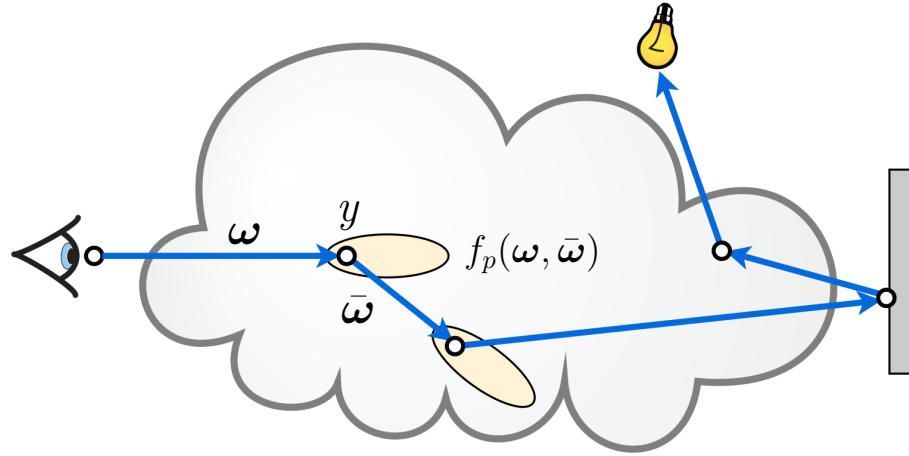


Fig. 2.17: Volumetric path tracing. *Image source:* [Now18]

Next event estimation (NEE)

When doing volumetric path tracing it is also possible and more efficient to connect directly to the light source at each bounce, this is known as *Next event estimation (NEE)*. We have said previously that when a light path does not end on the light source it is discarded as its contribution is zero. Instead of hoping that the path ends up on the light source, we can artificially connect to it using an efficient light source sampling strategy to compute the direct illumination part of scattering and then continue the path tracing process as before to compute the indirect illumination part of scattering independently.

Direction sampling

The random direction $\bar{\omega}$ is normally sampled by importance sampling the phase function:

$$p(\bar{\omega}) \propto f_p(\omega, \bar{\omega}). \quad (2.57)$$

If the phase function is isotropic then the direction needs to be sampled uniformly in a sphere and the PDF is the same as the phase function evaluation. In the anisotropic case (HG) the direction needs to be sampled using the scheme described below:

$$\begin{aligned} \cos \theta &= \frac{1}{2g} \left(1 + g^2 - \left(\frac{1 - g^2}{1 - g + 2g\xi} \right)^2 \right) \\ \phi &= 2\pi\xi. \end{aligned} \quad (2.58)$$

The PDF is computed exactly like the HG evaluation.

Distance sampling

The naive way is to uniformly sample a distance:

$$p(y) = \frac{1}{y_{max} - y_{min}}. \quad (2.59)$$

It is generally quite difficult to sample a distance within a medium by importance sampling the terms of the VRE. Assuming homogeneous media, a common strategy is to importance sample the transmittance from the camera:

$$p(y) \propto T(\mathbf{x}, \mathbf{y}) \quad (2.60)$$

$$p(y) \propto T(y). \quad (2.61)$$

Using the inversion method we can derive this sampling scheme:

$$y = -\frac{\ln(1 - \xi)}{\mu_t} \quad (2.62)$$

$$p(y) = \mu_t e^{-\mu_t y}. \quad (2.63)$$

CHAPTER 3

Related Work

In the past, most work has been focused on importance sampling point lights but not much addressed polygonal lights due to its inherent complexity. Rendering mesh lights efficiently proves to be even more challenging and is still subjects of ongoing research due to its recurrent use in the media and entertainment industry. In this section, we briefly present a few of those work from which we build our research on.

3.1 Equi-angular Sampling

We have seen in the previous chapter that a distance has be sampled inside the medium to obtain a correct rendering of scenes involving participating media. We have shown two different sampling strategies that are commonly used but it is also possible to sample this distance by importance sampling the geometry term of a point light, first proposed by Kulla et al. in their work *equi-angular sampling*[KF12]:

$$p(y) \propto \frac{1}{\|\mathbf{y} - \mathbf{p}\|^2}, \quad (3.1)$$

where $\mathbf{p}$ is the position of the point light. It is directly applicable to point lights and leads to substantial variance reduction for scenes with point light sources. This technique is also very effective for sphere lights and can even be applied to polygonal lights such as rectangular lights to considerably improve the rendering efficiency. However, this method completely ignores transmittance, as such it is significantly less effective in dense media.

3.1.1 Point Lights

Rendering of a point light within a single scattering homogeneous medium can be described using the area formulation of the VRE:

$$L(\mathbf{x}, \mathbf{z}) = \int_0^z T(\mathbf{x}, \mathbf{y}) \mu_s \int_{A_e} f_p(\mathbf{y}, \mathbf{x}, \mathbf{u}) T(\mathbf{y}, \mathbf{u}) L_e(\mathbf{y}, \mathbf{u}) V(\mathbf{y}, \mathbf{u}) G(\mathbf{y}, \mathbf{u}) dA(\mathbf{u}) dy. \quad (3.2)$$

Since a point light source has no surface, equation 3.2 reduces to:

$$L(\mathbf{x}, \mathbf{z}) = \mu_s \frac{\phi}{4\pi} \int_0^z T(\mathbf{x}, \mathbf{y}) f_p(\mathbf{y}, \mathbf{x}, \mathbf{p}) T(\mathbf{y}, \mathbf{p}) V(\mathbf{y}, \mathbf{p}) G(\mathbf{y}, \mathbf{p}) dy \quad (3.3)$$

$$G(\mathbf{y}, \mathbf{p}) = \frac{1}{\|\mathbf{y} - \mathbf{p}\|^2},$$

where ϕ is the emitted power of the point light. The geometry term can also be expressed in the angular frame first proposed by Lecocq et al.[LMAK00](See figure 3.1), where it becomes a function of two constants D and Δ :

$$G(\mathbf{y}, \mathbf{p}) = \frac{1}{D^2 + (y - \Delta)^2}. \quad (3.4)$$

The constant D is the distance between the point light position $\mathbf{p}$ and the projection of that point on the camera ray direction $\boldsymbol{\omega}$. The constant Δ is the distance between the camera ray origin $\mathbf{x}$ and the projected point.

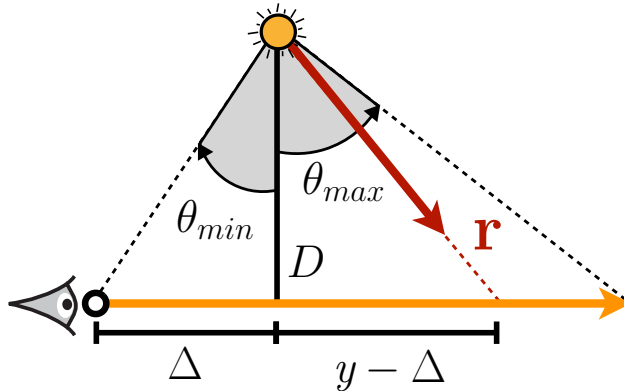


Fig. 3.1: Equi-angular sampling.

Deriving the scheme to importance sample this expression of the geometry term using the inversion method leads to an interesting geometrical interpretation: It is equivalent to sampling an angle θ uniformly between the angles bounding the camera ray $(\theta_{min}, \theta_{max})$ and projecting that angle back to a distance y on the camera ray. This angular distribution is commonly called *equi-angular*. The full sampling scheme is then:

$$\begin{aligned}
 \theta(\xi) &= (1 - \xi)\theta_{min} + \xi\theta_{max} \\
 p(\theta) &= \frac{1}{\theta_{max} - \theta_{min}} \\
 y(\xi) &= D \tan(\theta(\xi)) + \Delta \\
 J(\theta) &= \frac{D}{D^2 + y^2} \\
 p(y) &= p(\theta)J(\theta).
 \end{aligned} \tag{3.5}$$

On figure 3.2 we show results using this scheme to render a point light source in an homogeneous medium. Notice that equi-angular sampling generates a noise free image by only using 16 spp compared to transmittance sampling that would require many more samples.

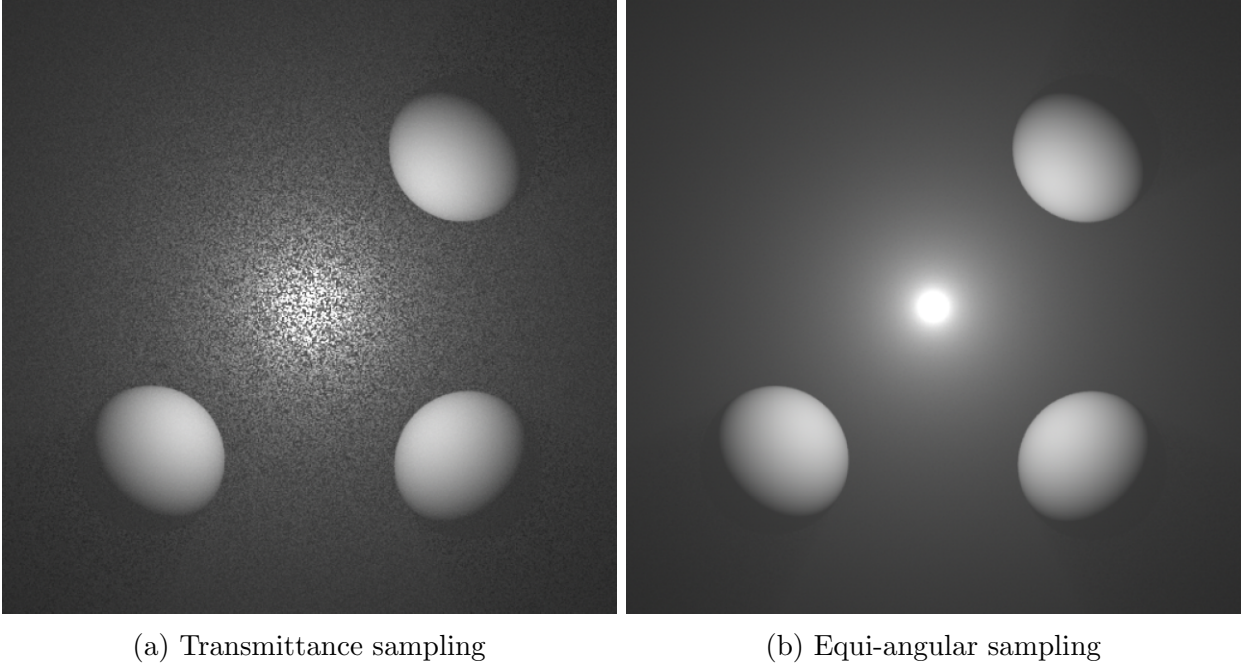


Fig. 3.2: Point light source in an homogeneous medium (16 spp). *Image source:* [KF12]

3.1.2 Sphere Lights

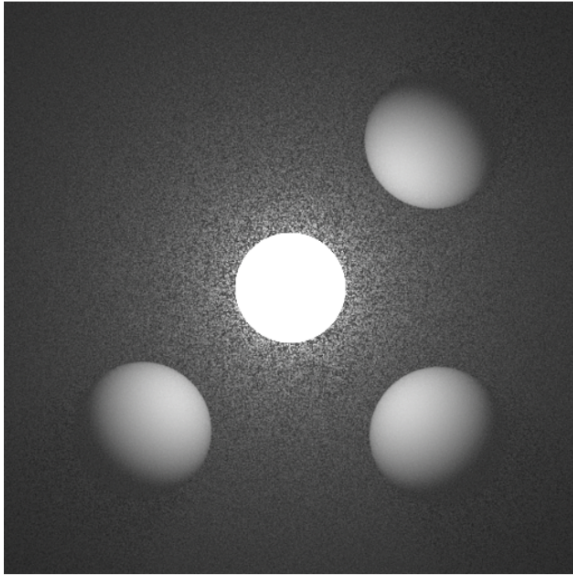
The illumination of a sphere light within a medium can be computed using the VRE:

$$L(\mathbf{x}, \boldsymbol{\omega}) = \int_0^z T(\mathbf{x}, \mathbf{y}) \mu_s \int_{S^2} f_p(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) T(\mathbf{y}, \mathbf{u}) L_e(\mathbf{u}, -\bar{\boldsymbol{\omega}}) d\bar{\boldsymbol{\omega}} d\mathbf{y}. \quad (3.6)$$

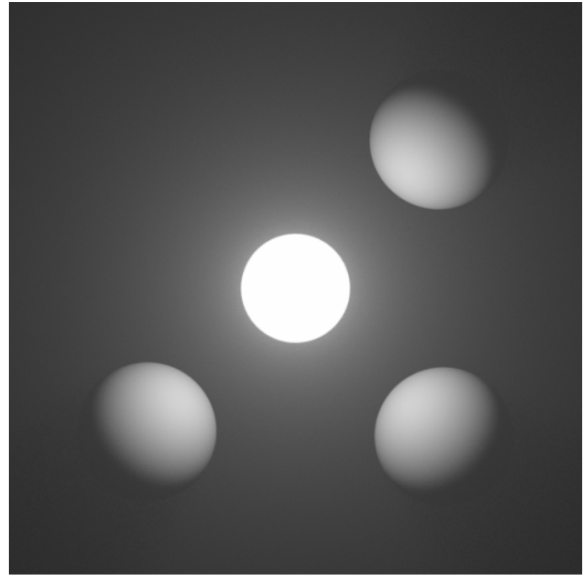
In this case we only consider single scattering homogeneous media and $L_e(\mathbf{u}, -\bar{\boldsymbol{\omega}})$ is from a sphere light source defined by its center $\mathbf{p}$ and emitted radiance L_e . This equation can be approximated using a Monte Carlo estimator:

$$L(\mathbf{x}, \boldsymbol{\omega}) \approx \frac{1}{N} \sum_{i=1}^N \frac{\mu_s T(\mathbf{x}, \mathbf{y}) f_p(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) T(\mathbf{y}, \mathbf{u}) L_e(\mathbf{u}, -\bar{\boldsymbol{\omega}})}{p(y)p(\bar{\boldsymbol{\omega}})}. \quad (3.7)$$

By using the method described in the work of Shirley et al.[SWZ96], the random direction can be sampled proportionally to the subtended solid angle of the sphere light $\Omega_e(y)$ such that $p(\bar{\boldsymbol{\omega}}) \propto \Omega_e(y)$. The best way to sample the distance would thus be $p(y) \propto \Omega_e(y)$. No importance sampling scheme for this case has been derived yet but equi-angular sampling provides a very good alternative as we can see in figure 3.3 since $\Omega_e(y) \propto \frac{1}{\|\mathbf{y}-\mathbf{p}\|^2}$ for sphere lights.



(a) Transmittance sampling



(b) Equi-angular sampling

Fig. 3.3: Sphere light source in an homogeneous medium (16 spp). *Image source:* [KF12]

3.1.3 Polygonal Lights

Equi-angular sampling can also be used for polygonal lights but its efficiency is vastly reduced as a polygonal light differs widely from a point or sphere light due to its illumination being concentrated in the direction of its surface's normal. Nevertheless, by placing an imaginary point light at the center of the polygonal light and importance sampling its geometry term using equi-angular sampling is more efficient than doing uniform sampling or importance sampling only the transmittance of the camera. It is also relatively easy to importance sample the clamping of the cosine from the geometry term of a polygonal light in addition to the geometry term of the imaginary point light, reducing the variance a bit more. As we

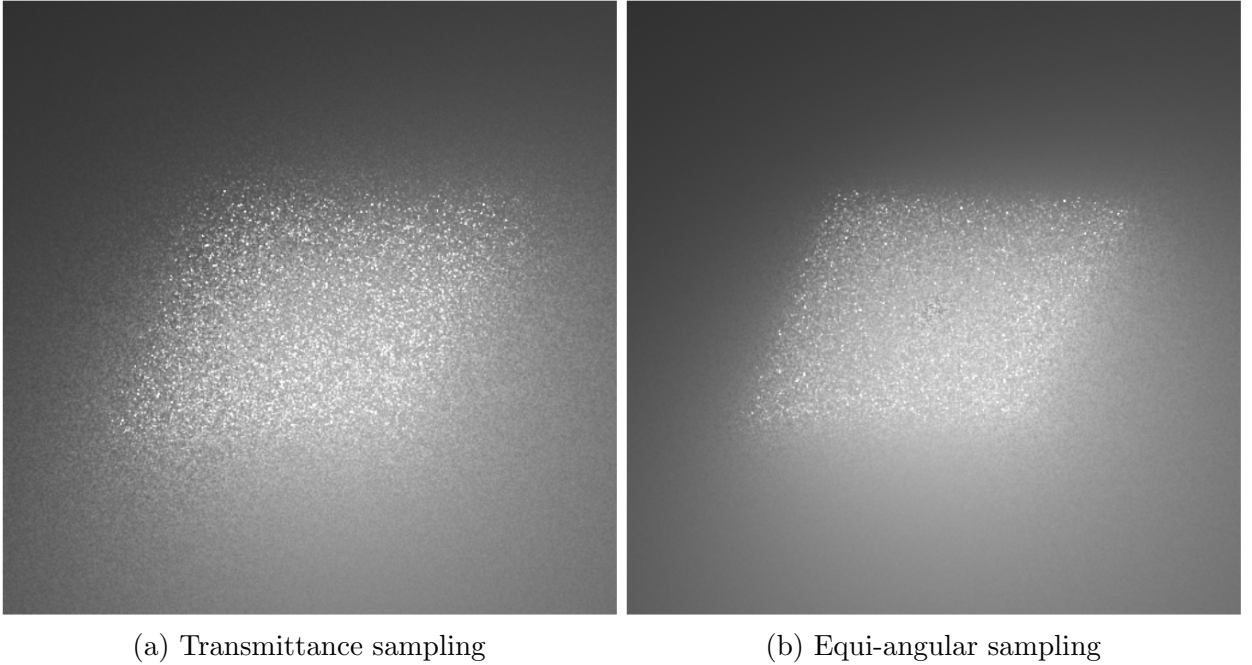


Fig. 3.4: Rectangular light source in an homogeneous medium (256 spp). *Image source:* [KF12]

can see in figure 3.4 equi-angular sampling provides a substantial improvement compared to transmittance sampling but the image still stays very noisy even at 256 spp. Most of the noise left is located around the light source due to the fact that equi-angular sampling does not take into account the surface's normal of the light source and since the illumination is spread out along the area of the light. This is what we attempt to rectify in our work.

3.1.4 Transmittance MIS

Since it does not seem to be possible to importance sample the product of the transmittance from the camera $T(\mathbf{x}, \mathbf{y})$ and the geometry term of a point light, equi-angular sampling[KF12] proposes to use MIS instead. This scheme is evidently non-optimal but results in better variance reduction than only importance sampling one of the term and it is simple to implement. Including the transmittance from the light $T(\mathbf{y}, \mathbf{p})$ proves to be even more difficult due to the resulting form of the mathematical expression necessary to define the distance between $\mathbf{y}$ and $\mathbf{p}$ as a function of the sampled distance y ,

$$T(\mathbf{y}, \mathbf{p}) = e^{-\mu_t \sqrt{D^2 + (y - \Delta)^2}}. \quad (3.8)$$

Importance sampling only the transmittance from the light is not even possible since no analytical solution exists for

$$\int_0^z e^{-\mu_t \sqrt{D^2 + (y - \Delta)^2}} dy. \quad (3.9)$$

3.2 Joint Importance Sampling (JIS)

Follow up work to equi-angular sampling is *Joint Importance Sampling (JIS)* by Georgiev et al.[GKH⁺13]. This work derives an importance sampling scheme for the product of the geometry term (G) of a point light and the phase function (f_p) for multiple scattering media. Note that this work ignores the transmittance and the visibility terms. Given a light path $\bar{s}$ of two bounces in an homogeneous scattering medium (Shown in orange on figure 3.5), its

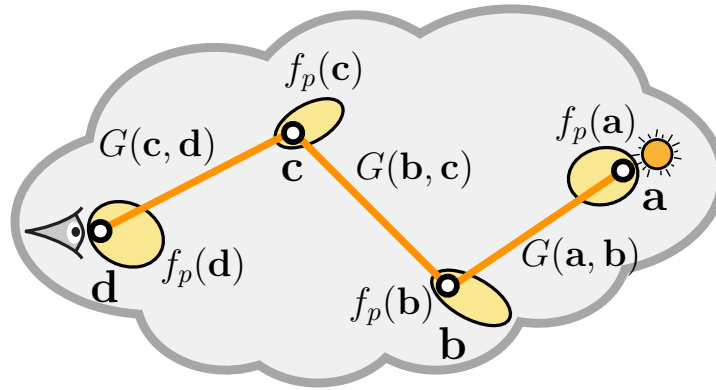


Fig. 3.5: Light path of two bounces in an homogeneous scattering medium.

Monte Carlo estimator can be defined as:

$$L(\mathbf{x}, \boldsymbol{\omega}) \approx \frac{1}{N} \sum_{i=1}^N \frac{f(\bar{s})}{p(\bar{s})}, \quad (3.10)$$

where $f(\bar{s})$ is the full contribution of the light path $\bar{s}$. The best way to importance sample this path is by sampling a path such that the $p(\bar{s}) \propto p(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$, where the joint PDF $p(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$ is defined to be $f_p(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})G(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$. This joint PDF is normally sampled using local sampling PDFs such as

$$p(\mathbf{d})p(\mathbf{c}|\mathbf{d})p(\mathbf{b}|\mathbf{c})p(\mathbf{a}) \rightarrow p(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}). \quad (3.11)$$

This work however proposes to sample from the joint PDF directly instead, from which the local sampling PDFs are then derived:

$$p(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \rightarrow p(\mathbf{d})p(\mathbf{c}|\mathbf{d})p(\mathbf{b}|\mathbf{c})p(\mathbf{a}). \quad (3.12)$$

Joint importance sampling relies on the key observation that in the case of rendering in participating media it is possible to prescribe this joint PDF since, contrary to surface rendering, the path is not constrained to a surface and as such can be freely moved inside the medium. While their work shows how to importance sample a path of only two bounces, it can theoretically be generalized to paths of any length. Since we focus on single scattering, the part that is the most interesting to us is the sampling scheme to importance sample the product of the geometry term and the phase function. The key insight here is to realize

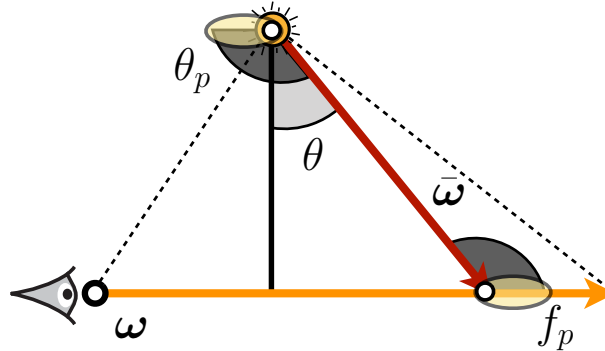


Fig. 3.6: Anisotropic phase function in the equi-angular frame.

that the phase function's angle θ_p can be trivially mapped back to the equi-angular frame by offsetting the angle by $\pi/2$ (See figure 3.6). In this frame the importance sampling scheme can be derived using the inversion method. Finally, joint importance sampling handles transmittance similarly to equi-angular sampling by using MIS.

3.3 Lightcuts

It is quite challenging to render scenes with mesh lights since they can essentially be seen as a large number of polygonal (triangle) lights. The problem of rendering with mesh lights becomes then a problem of rendering many small polygonal lights very efficiently. An interesting thing to notice is that for most scenes a great number of those triangle lights will not contribute much or at all if the light is backfacing the camera. It thus makes sense to develop a method to group them according to their contribution and sample more often the ones that contribute much more than the rest. One pioneer work on this problem is *Lightcuts* by Walter et al. [WFA⁺05]. Lightcuts has a lot more applications than rendering mesh lights, but we will not cover them as they are not of interest to our work. Furthermore, it was only applied to surface rendering and as such it is not possible to render participating media using it.

Lightcuts proposes to cluster the lights using a similarity metric based on their position and direction. The clusters are then stored in a hierarchical data structure (binary tree) for fast retrieval when rendering (See figure 3.7). At rendering, the binary tree is traversed to only select a small subset of the light sources that contribute significantly to the given shading point. This small subset is called a *lightcut*, which is fundamentally a specific clustering of the light sources that depends on the shading point (See figure 3.8). An iterative algorithm is used to refine the lightcut based on the maximum error bound of the illumination. The cut is refined as long as the error is higher than 2% of the difference between the real illumination and the approximation (error bound) i.e. if the difference is perceptible. The resulting algorithm is sub-linear on the number of triangle lights forming the mesh light.

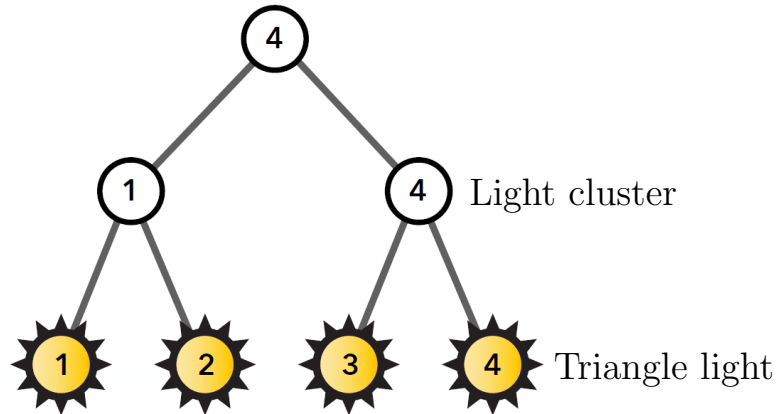


Fig. 3.7: Example of a binary tree used to cluster the triangle lights forming a mesh light in Lightcuts. *Image source:* [Now18]

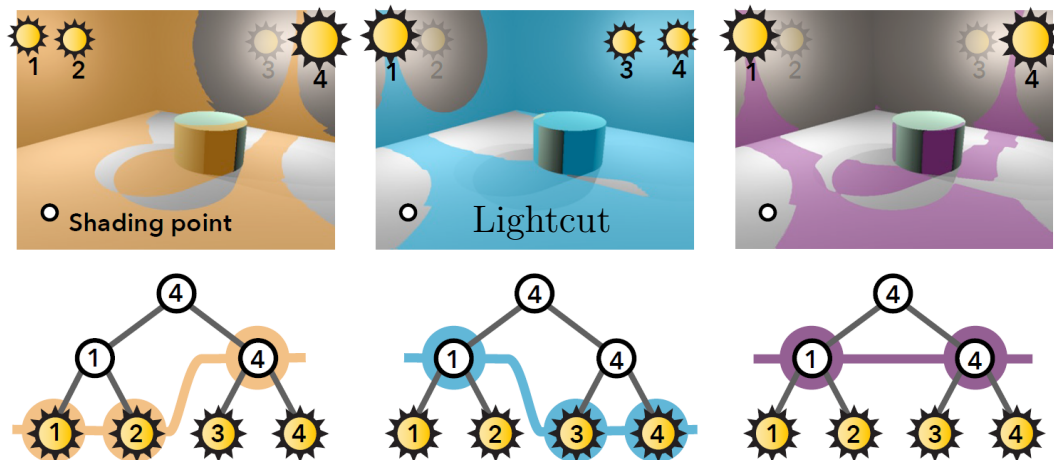


Fig. 3.8: A shading point under different lightcuts and the corresponding approximation to its illumination. *Image source:* [Now18]

CHAPTER 4

Importance Sampling Polygonal Lights

Rendering polygonal lights in participating media requires to solve the volume rendering equation. This equation involves many terms of different nature such as: the geometry term, the transmittance and the phase function. Due to their distinct structure, we show how to importance sample each of those term by studying them one at a time and extending our sampling scheme accordingly. We will see that since importance sampling those terms for the full polygonal light does not seem to be possible, we propose to importance sample them for a small set of oriented point light sources on the surface of a polygonal light. Finally, we show ongoing work to extend this sampling scheme to mesh lights using a hierarchical light clustering strategy.

4.1 Polygonal Lights in Participating Media

The volume rendering equation (equation 2.37) can be used to render polygonal lights in a medium. In our work we only consider the case of a scattering medium, ignoring the absorption and emission of light within the medium by considering that the absorption coefficient μ_a is zero. These assumptions lead to the following formulation of the VRE:

$$L(\mathbf{x}, \boldsymbol{\omega}) = \int_0^z T(\mathbf{x}, \mathbf{y}) \mu_s(\mathbf{y}) \int_{S^2} f_p(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) L_i(\mathbf{y}, \bar{\boldsymbol{\omega}}) d\bar{\boldsymbol{\omega}} dy. \quad (4.1)$$

To reduce the scope of the problem we focus on single scattering by considering that $L_i(\mathbf{y}, \bar{\omega})$ comes directly from the light source. The resulting equation is then:

$$L(\mathbf{x}, \omega) = \int_0^z T(\mathbf{x}, \mathbf{y}) \mu_s(\mathbf{y}) \int_{S^2} f_p(\omega, \bar{\omega}) T(\mathbf{y}, \mathbf{u}) L_e(\mathbf{u}, -\bar{\omega}) d\bar{\omega} dy, \quad (4.2)$$

where $\mathbf{u}$ is a position on the light source and L_e is coming from one flat polygonal light source such as a rectangular light. We simplify the problem further by only considering an infinite homogeneous medium so that the transmittance expression is simplified and the μ_s and μ_t coefficients become spatially invariant (See figure 4.1). The VRE under those conditions is thus:

$$L(\mathbf{x}, \omega) = \int_0^z T(\mathbf{x}, \mathbf{y}) \mu_s \int_{S^2} f_p(\omega, \bar{\omega}) T(\mathbf{y}, \mathbf{u}) L_e(\mathbf{u}, -\bar{\omega}) d\bar{\omega} dy. \quad (4.3)$$

This equation can equivalently be expressed in the area formulation:

$$L(\mathbf{x}, \omega) = \int_0^z T(\mathbf{x}, \mathbf{y}) \mu_s \int_{A_e} f_p(\omega, \bar{\omega}) T(\mathbf{y}, \mathbf{u}) L_e(\mathbf{u}, -\bar{\omega}) dA(\mathbf{u}) dy, \quad (4.4)$$

to reveal all the terms it contains and to show how challenging it is to importance sample.

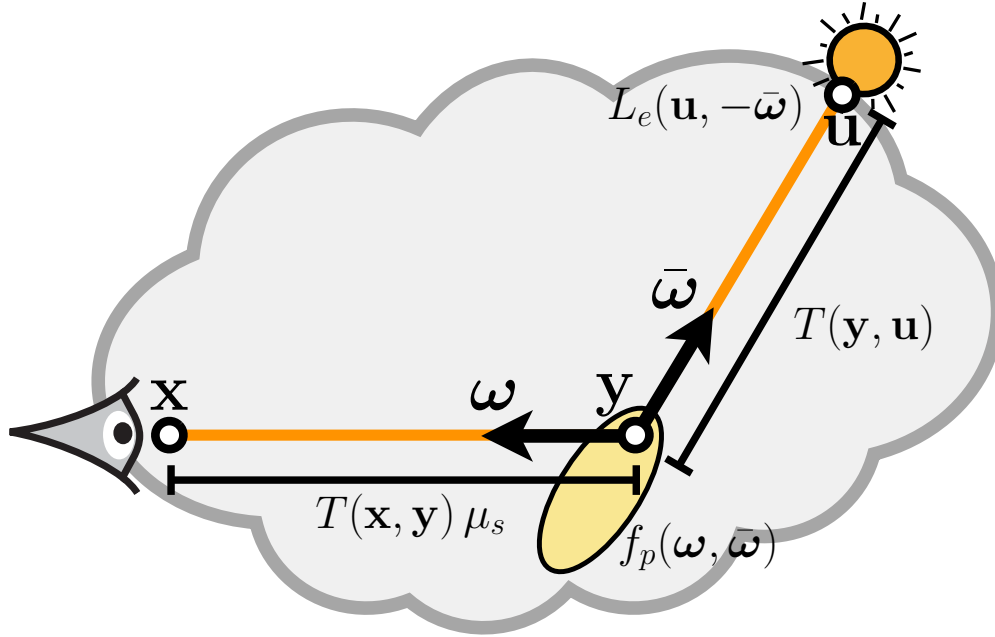


Fig. 4.1: Single scattering VRE.

To efficiently render polygonal lights in participating media using a Monte Carlo estimator such as:

$$L(\mathbf{x}, \omega) \approx \frac{1}{N} \sum_{i=1}^N \frac{T(\mathbf{x}, \mathbf{y}) \mu_s f_p(\mathbf{y}, \mathbf{x}, \mathbf{u}) T(\mathbf{y}, \mathbf{u}) L_e(\mathbf{y}, \mathbf{u}) V(\mathbf{y}, \mathbf{u}) G(\mathbf{y}, \mathbf{u})}{p(y)p(\bar{\omega})}, \quad (4.5)$$

we ideally should importance sample all the terms of equation 4.4:

- Geometry: $G(\mathbf{y}, \mathbf{u})$
- Transmittance from the camera: $T(\mathbf{x}, \mathbf{y})$
- Transmittance from the light: $T(\mathbf{y}, \mathbf{u})$
- Phase function: $f_p(\mathbf{y}, \mathbf{x}, \mathbf{u})$

We ignore the visibility term $V(\mathbf{y}, \mathbf{u})$ as it is usually not possible to importance sample. The geometry term defines the illumination produced by the polygonal light source depending on the shape of the light, its position and the direction of its normal. Both of the transmittance terms arise from the same phenomenon of light attenuation within the medium. The only difference between the two terms is the part of the light path where the attenuation takes place. The transmittance from the camera is the attenuation along the ray from the camera origin $\mathbf{x}$ to the shading point $\mathbf{y}$, while the transmittance from the light is the attenuation between the shading point $\mathbf{y}$ and a point $\mathbf{u}$ on the light source. Lastly, the phase function defines how the medium scatters light according to a given probability distribution. It follows that since each of those term comes from a different physical process they behave very differently and thus need to be handled individually.

4.2 Geometry term

We first show how to importance sample the geometry term separately from all the other terms as it is the only term that truly defines the *shape* of the illumination from a polygonal light source and all the other terms indirectly depend on it. We then build up on this method to importance sample the product of the geometry term with the other terms.

4.2.1 Polygonal Light

The geometry term of a polygonal light is defined as:

$$G(\mathbf{y}, \mathbf{u}) = \frac{|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|}{\|\mathbf{y} - \mathbf{u}\|^2}, \quad (4.6)$$

where $n(\mathbf{u})$ is the light's normal at the position $\mathbf{u}$. By temporarily ignoring the phase function and the transmittance terms to focus only on the geometry term, equation 4.4 becomes:

$$L(\mathbf{x}, \mathbf{z}) = \mu_s \int_0^z \int_{A_e} L_e(\mathbf{y}, \mathbf{u}) \frac{|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|}{\|\mathbf{y} - \mathbf{u}\|^2} dA(\mathbf{u}) dy. \quad (4.7)$$

If we use the usual directional formulation of equation 4.7 we get:

$$L(\mathbf{x}, \boldsymbol{\omega}) = \mu_s \int_0^z \int_{S^2} L_e(\mathbf{u}, -\bar{\boldsymbol{\omega}}) d\bar{\boldsymbol{\omega}} dy. \quad (4.8)$$

Since the direction integral only contributes in the region that is subtended by the light source, we only consider this region Ω which is equivalent to computing the solid angle subtended by the light source Ω_e and parametrize it as a function of distance y :

$$L(\mathbf{x}, \boldsymbol{\omega}) = \mu_s L_e \int_0^z \int_{\Omega} d\bar{\boldsymbol{\omega}} dy = \mu_s L_e \int_0^z \Omega_e(y) dy. \quad (4.9)$$

Given its Monte Carlo estimator,

$$L(\mathbf{x}, \boldsymbol{\omega}) \approx \frac{1}{N} \sum_{i=1}^N \frac{\mu_s L_e \Omega_e(y)}{p(y)}, \quad (4.10)$$

the way to perfectly importance sample this equation would be:

$$p(y) \propto \Omega_e(y). \quad (4.11)$$

The subtended solid angle of a flat polygonal shape Ω_e can be computed using the equation from[Arv95]:

$$\begin{aligned} \Omega_e &= \alpha + \beta + \gamma - \pi \\ \alpha &= \arccos \frac{(\mathbf{B} \times \mathbf{A}) \cdot (\mathbf{A} \times \mathbf{C})}{\|\mathbf{B} \times \mathbf{A}\| \|\mathbf{A} \times \mathbf{C}\|} \\ \beta &= \arccos \frac{(\mathbf{C} \times \mathbf{B}) \cdot (\mathbf{B} \times \mathbf{A})}{\|\mathbf{C} \times \mathbf{B}\| \|\mathbf{B} \times \mathbf{A}\|} \\ \gamma &= \arccos \frac{(\mathbf{A} \times \mathbf{C}) \cdot (\mathbf{C} \times \mathbf{B})}{\|\mathbf{A} \times \mathbf{C}\| \|\mathbf{C} \times \mathbf{B}\|}. \end{aligned} \quad (4.12)$$

Here shown for a triangle (See figure 4.2), but it is trivial to generalize it to any polygon. This equation assumes that the subtended solid angle is computed at the origin $(0, 0, 0)$, but it is easy to parametrize it as a function of distance into the medium.

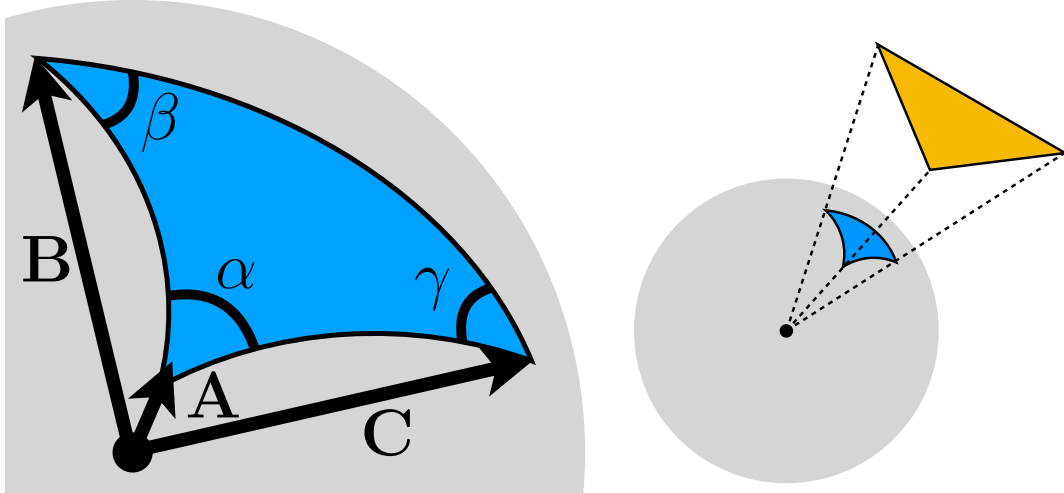


Fig. 4.2: The subtended solid angle of a triangular light (yellow) is the area of the projection of the triangle onto a unit sphere (blue), or equivalently, the sum of the angles of the resulting spherical triangle offsetted by π .

To derive the importance sampling scheme for equation 4.10 using the inversion method, the CDF:

$$P(y) = \int_0^z \Omega_e(y) dy, \quad (4.13)$$

needs to be calculated. Unfortunately an analytical solution to this integral does not seem to be possible. As such, we express a polygonal light as M *point-normal* lights at the surface of the polygonal light like seen on figure 4.3. The illumination of a polygonal light (equation 4.7) can thus be seen as an approximation of M point-normal lights:

$$\mu_s \int_0^z \int_{A_e} L_e(\mathbf{y}, \mathbf{u}) \frac{|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|}{\|\mathbf{y} - \mathbf{u}\|^2} dA(\mathbf{u}) dy \approx \mu_s L_e \sum_{k=1}^M \int_0^z \frac{|n(\mathbf{u}_k) \cdot (\mathbf{y} - \mathbf{u}_k)|}{\|\mathbf{y} - \mathbf{u}_k\|^2} dy, \quad (4.14)$$

where $\mathbf{u}_k$ is the position of the k^{th} point-normal light. The point-normal (PN) is a theoretical light composed of a point light that emits only in one direction by its cosine lobe. It is fully defined by its position $\mathbf{u}$ and normal $n(\mathbf{u})$. To importance sample a polygonal light we then randomly choose one PN out of the M point-normal lights and importance sample the geometry term of the chosen PN. We will go into more details on how to choose this PN later on.

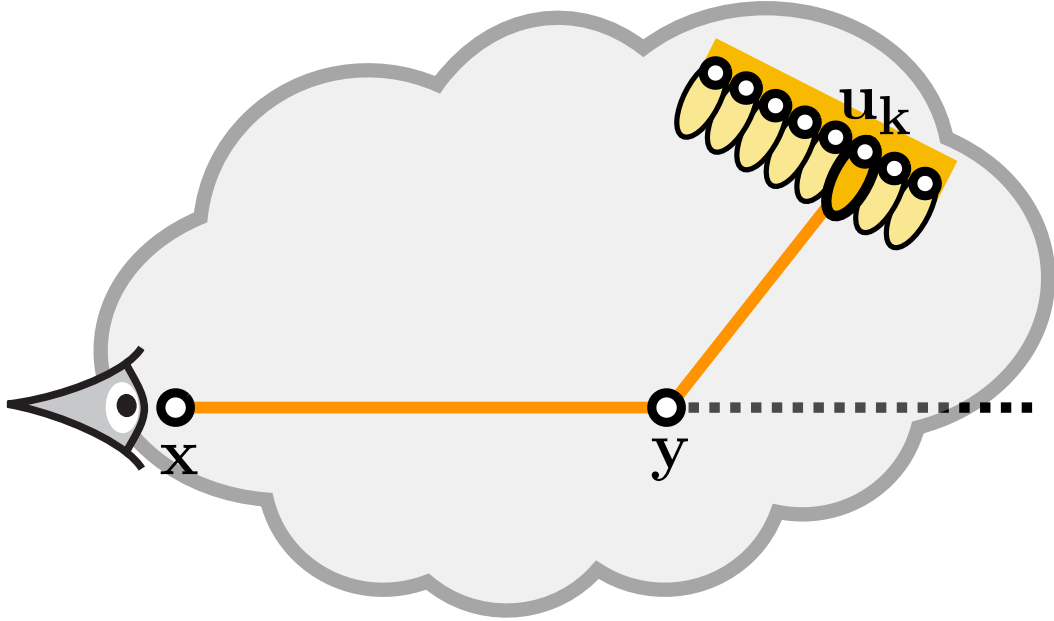


Fig. 4.3: Approximation of a polygonal light using point-normal lights.

4.2.2 Point-normal Light

Given the illumination of a single point-normal in a medium:

$$L(\mathbf{x}, \mathbf{z}) = \int_0^z \frac{|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|}{\|\mathbf{y} - \mathbf{u}\|^2} dy = \int_0^z \frac{|n(\mathbf{u}) \cdot r(y)|}{\|r(y)\|^2} dy, \quad (4.15)$$

we can approximate this integral using a Monte Carlo estimator:

$$\int_0^z \frac{|n(\mathbf{u}) \cdot r(y)|}{\|r(y)\|^2} dy \approx \frac{1}{N} \sum_{i=1}^N \frac{|n(\mathbf{u}) \cdot r(y)|}{\|r(y)\|^2} \frac{1}{p(y)}. \quad (4.16)$$

To perfectly importance sample this equation we need to define the PDF as

$$p(y) \propto \frac{|n(\mathbf{u}) \cdot r(y)|}{\|r(y)\|^2}. \quad (4.17)$$

Angular domain

Inspired by equi-angular sampling[KF12], we convert this distance integral to the angular domain by performing a change of variable:

$$\begin{aligned} y(\theta) &= D \tan \theta + \Delta \\ J(\theta) &= \frac{d(y(\theta))}{d\theta} = \frac{D}{\|r(\theta)\|^2} \end{aligned} \quad (4.18)$$

$$\int_0^z \frac{|n(\mathbf{u}) \cdot r(y)|}{\|r(y)\|^2} dy = \int_{\theta_1}^{\theta_2} \frac{|n(\mathbf{u}) \cdot r(\theta)|}{\|r(\theta)\|^2} J(\theta) d\theta \approx \frac{1}{N} \sum_{i=1}^N \frac{|n(\mathbf{u}) \cdot r(\theta)|}{\|r(\theta)\|^2} J(\theta) \frac{1}{p(\theta)}. \quad (4.19)$$

If we importance sample the $|n(\mathbf{u}) \cdot r(\theta)|$ term in equation 4.19, then the PDF needs to be:

$$p(\theta) = \frac{|n(\mathbf{u}) \cdot r(\theta)|}{\int_{\theta_1}^{\theta_2} |n(\mathbf{u}) \cdot r(\theta)| d\theta}. \quad (4.20)$$

If we then go back to the distance domain and we use $p(y) = p(\theta)J(\theta)$:

$$\int_0^z \frac{|n(\mathbf{u}) \cdot r(y)|}{\|r(y)\|^2} dy \approx \frac{1}{N} \sum_{i=1}^N \frac{|n(\mathbf{u}) \cdot r(\theta)|}{\|r(\theta)\|^2} \frac{1}{\frac{\int_{\theta_1}^{\theta_2} |n(\mathbf{u}) \cdot r(\theta)| d\theta}{\|r(\theta)\|^2} \frac{D}{\|r(\theta)\|^2}}, \quad (4.21)$$

every term cancel out except the proportionality constant $\frac{1}{D}$. This means that by going to the angular domain we essentially get the $\frac{1}{\|r(y)\|^2}$ importance sampled for free:

$$\int_0^z \frac{|n(\mathbf{u}) \cdot r(y)|}{\|r(y)\|^2} dy = \frac{\int_{\theta_1}^{\theta_2} |n(\mathbf{u}) \cdot r(\theta)| d\theta}{D}. \quad (4.22)$$

Thus by sampling an angle θ proportionally to $|n(\mathbf{u}) \cdot r(\theta)|$ and projecting that angle to the corresponding distance on the camera ray (See figure 4.4), we obtain a distance sampled proportionally to the full geometry term of a point-normal:

$$p(y) \propto \frac{|n(\mathbf{u}) \cdot r(y)|}{\|r(y)\|^2}. \quad (4.23)$$

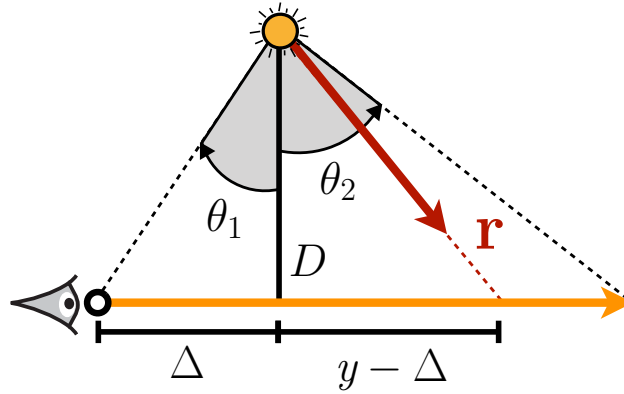


Fig. 4.4: Projection of an angle to a distance in the equi-angular frame.

Clamping

To compute the integral from equation 4.22:

$$\int_{\theta_1}^{\theta_2} |n(\mathbf{u}) \cdot r(\theta)| d\theta, \quad (4.24)$$

we need to explicitly handle the clamping of $|n(\mathbf{u}) \cdot r(\theta)|$. Any $r(\theta)$ sampled behind the point-normal's plane will have a contribution of zero since $r(\theta) \cdot n(\mathbf{u}) < 0$. On figure 4.5 we observe that the effect of clamping only changes the boundaries of integration. We thus compute the new integration boundaries θ_{min} and θ_{max} by figuring out the angle θ_{hit} at the point of intersection of the point-normal's plane with the camera ray. We also need to take into account the normal's orientation relative to the ray. There are essentially four cases:

$$(\theta_{min}, \theta_{max}) = \begin{cases} (\theta_1, \theta_2) & \text{Contribute everywhere} \\ (\theta_{hit}, \theta_2) & \text{Contribute partially and same orientation} \\ (\theta_1, \theta_{hit}) & \text{Contribute partially and opposite orientation} \\ (0, 0) & \text{Contribute nowhere} \end{cases} \quad (4.25)$$

The original integral can then be simplified to:

$$\int_{\theta_1}^{\theta_2} |n(\mathbf{u}) \cdot r(\theta)| d\theta = \int_{\theta_{min}}^{\theta_{max}} n(\mathbf{u}) \cdot r(\theta) d\theta. \quad (4.26)$$

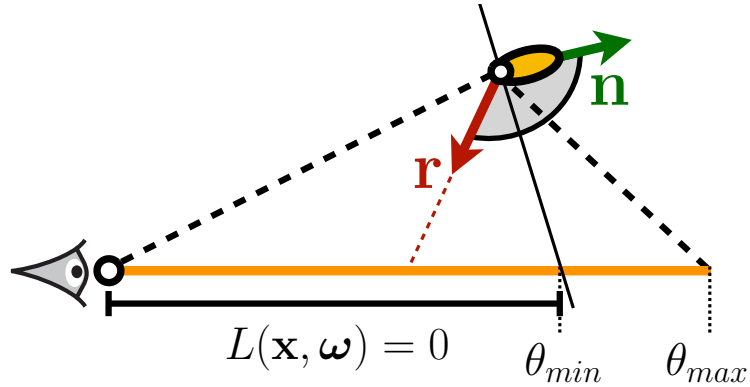


Fig. 4.5: Clamping the CDF only results in a change of the integration boundaries.

Parametrization of $r(\theta)$

The sweeping vector r is parametrized as a function of an angle θ in polar coordinates:

$$r(\theta) = (\cos \theta, \sin \theta). \quad (4.27)$$

There are many possible ways to reference this angle. The natural frame (figure 4.6b) seems convenient but we propose to use the equi-angular frame. In this frame (figure 4.6a), the angle θ is referenced from D and varies between $(-\pi/2, \pi/2)$. The angles on the left side of D are always negative and the angles on the right side are always positive. The constant Δ and D are defined exactly like equi-angular sampling. While using the equi-angular frame results in a more complex PDF and CDF expression, it simplifies the projection of the angular samples to the distance samples and also helps later on when parametrizing the transmittance. Additionally, it drastically simplifies the implementation of our sampling scheme into existing renderers as it is using the same frame as equi-angular sampling.

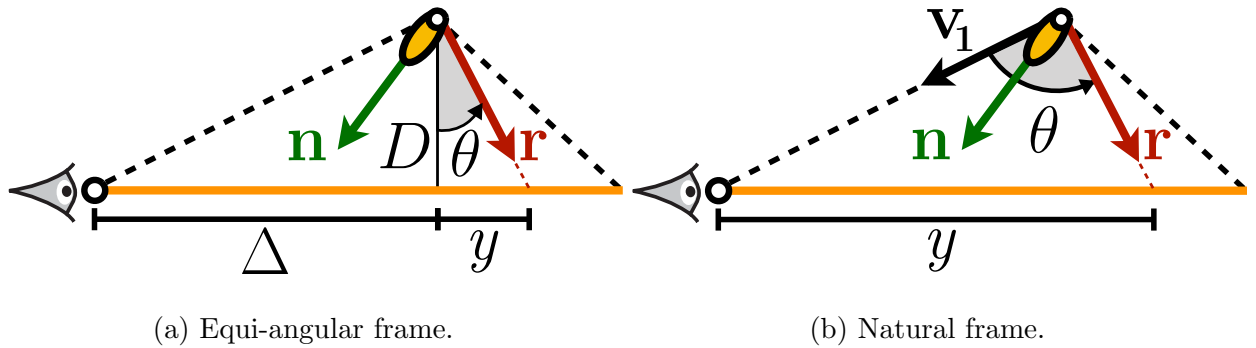


Fig. 4.6: Angular frames.

Parametrization of $n(\mathbf{u})$

The sweeping vector $r(\theta)$ is defined in the 2D plane formed by the camera ray and the point-normal light but the PN's normal $n(\mathbf{u})$ is defined in the 3D world space of the scene. Since both need to be in the same frame of reference to be able to compute the cosine $r(\theta) \cdot n(\mathbf{u})$ correctly, we transform the normal into the same 2D plane as the sweeping vector using the frame $\mathbf{F}$ as seen on figure 4.7. The frame $\mathbf{F}$ is computed using:

$$\begin{aligned}\mathbf{F} &= [\mathbf{D} \quad \mathbf{f} \quad \mathbf{z}] \\ \mathbf{z} &= \mathbf{v}_1 \times \mathbf{D} \\ \mathbf{f} &= \mathbf{z} \times \mathbf{D}.\end{aligned}\tag{4.28}$$

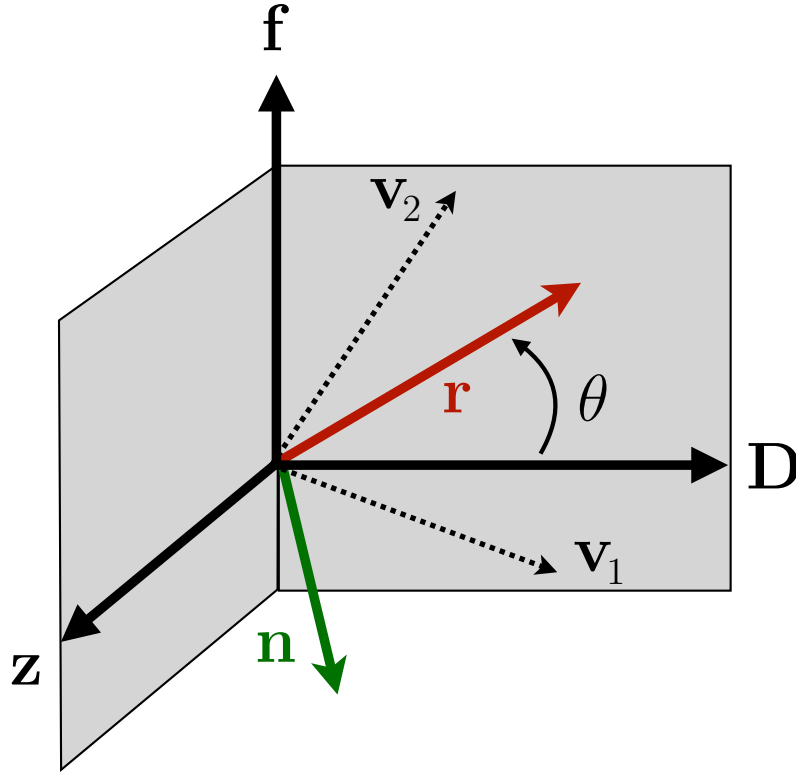


Fig. 4.7: Frame for the normal.

The expression to transform the normal into this frame is thus:

$$n_F(\mathbf{u}) = n(\mathbf{u}) \mathbf{F}^{-1} = \begin{bmatrix} n(\mathbf{u}) \cdot D \\ n(\mathbf{u}) \cdot \mathbf{f} \\ n(\mathbf{u}) \cdot \mathbf{z} \end{bmatrix}. \quad (4.29)$$

Since $r(\theta)$ is 2D and we are only interested in computing the dot product, we can ignore $\mathbf{F}_z^{-1}$. We refer to the x component of $n_F(\mathbf{u})$ as A and the y component as B , since they are constant for a given ray. The special case where Δ is negative also needs to be taken into account by inverting the $\mathbf{f}$ vector of the frame. The resulting expression for both constants is then:

$$\begin{aligned} A &= n(\mathbf{u}) \cdot D \\ B &= \text{sgn}(\Delta) n(\mathbf{u}) \cdot \mathbf{f}. \end{aligned} \quad (4.30)$$

PDF normalization

The angular PDF is computed as:

$$p(\theta) = c(n_F(\mathbf{u}) \cdot r(\theta)) = c((A, B) \cdot (\cos \theta, \sin \theta)) = c(A \cos \theta + B \sin \theta). \quad (4.31)$$

The normalization factor c is computed as:

$$\begin{aligned} c &= \frac{1}{\int_{\theta_{min}}^{\theta_{max}} (A \cos \theta + B \sin \theta) d\theta} \\ &= \frac{1}{(A \sin \theta - B \cos \theta) \Big|_{\theta_{min}}^{\theta_{max}}} \\ &= \frac{1}{A(\sin \theta_{max} - \sin \theta_{min}) - B(\cos \theta_{max} - \cos \theta_{min})}. \end{aligned} \quad (4.32)$$

The full normalized PDF and CDF expressions are then:

$$p(\theta) = \frac{A \cos(\theta) + B \sin(\theta)}{A(\sin \theta_{max} - \sin \theta_{min}) - B(\cos \theta_{max} - \cos \theta_{min})} \quad (4.33)$$

$$P(\theta) = \frac{A(\sin \theta - \sin \theta_{min}) - B(\cos \theta - \cos \theta_{min})}{A(\sin \theta_{max} - \sin \theta_{min}) - B(\cos \theta_{max} - \cos \theta_{min})}. \quad (4.34)$$

Sampling

To sample an angle from the PDF, the normalized CDF needs to be inverted:

$$\theta(\xi) = P^{-1}(\xi). \quad (4.35)$$

Since the normalization factor is a constant, the constants A and B can be adjusted to take it into account and only invert the unnormalized CDF. Those normalized constants are referred to A_* and B_* respectively, leading to the following formulation:

$$\xi = P(\theta) = A_*(\sin \theta - \sin \theta_{min}) - B_*(\cos \theta - \cos \theta_{min}). \quad (4.36)$$

Moving constants to one side of the equation and variables on the other:

$$\xi + A_* \sin \theta_{min} - B_* \cos \theta_{min} = A_* \sin \theta - B_* \cos \theta. \quad (4.37)$$

By solving for θ and grouping common terms in the expression we get:

$$\begin{aligned} U &= A_* \sin \theta_{min} - B_* \cos \theta_{min} \\ V &= \sqrt{A_*^2 + B_*^2 - (\xi + U)^2} \\ Q &= A_*(\xi + U) \\ R &= \text{sgn}(A_*)B_* V \\ S &= -B_*(\xi + U) \\ T &= V |A_*| \\ \theta(\xi) &= \arctan \left(\frac{Q \pm R}{S \pm T} \right). \end{aligned} \quad (4.38)$$

This equation has two solutions, but for any given situation only one of them will give a θ between θ_{min} and θ_{max} . To sample a distance y the sampled angle θ needs to be projected back to the distance domain and the PDF needs to be adjusted using the Jacobian $J(\theta)$.

This projection is the same as equi-angular sampling:

$$y(\xi) = D \tan(\theta(\xi)) + \Delta \quad (4.39)$$

$$J(\theta) = \frac{D}{D^2 + y^2} \quad (4.40)$$

$$p(y) = p(\theta)J(\theta). \quad (4.41)$$

The only difference between our sampling scheme and equi-angular sampling is that we sample an angle proportionally to the cosine while equi-angular sampling samples an angle uniformly.

4.3 Transmittance

We now show how to extend our sampling scheme to importance sample the product of the geometry term with the transmittance of both the camera and the light. We will see that since we are unable to find an analytical solution to include the transmittance terms we use an approximation instead.

4.3.1 Analytical Solution

Importance sampling the product of the geometry term and the transmittance terms $T(\mathbf{x}, \mathbf{y})$ and $T(\mathbf{y}, \mathbf{u})$ is quite a bit more challenging since the integrand now includes the transmittance terms which is an exponential decay in function of the distance that we are trying to sample:

$$L(\mathbf{x}, \mathbf{z}) = \mu_s L_e \int_0^z T(\mathbf{x}, \mathbf{y}) T(\mathbf{y}, \mathbf{u}) \frac{|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|}{\|\mathbf{y} - \mathbf{u}\|^2} dy. \quad (4.42)$$

We express this equation as a function of distance to reveal the relationship to the sampling distance y :

$$L(\mathbf{x}, \mathbf{z}) = \mu_s L_e \int_0^z T_C(y) T_L(y) \frac{|n(\mathbf{u}) \cdot r(y)|}{\|r(y)\|^2} dy, \quad (4.43)$$

where the transmittance from the camera is:

$$T_C(y) = e^{-\mu_t y}, \quad (4.44)$$

and the transmittance from the light is:

$$T_L(y) = e^{-\mu_t \sqrt{D^2 + (y - \Delta)^2}}. \quad (4.45)$$

By expressing equation 4.43 in the angular domain similarly to what we have done previously for the geometry term we obtain:

$$L(\mathbf{x}, \mathbf{z}) = \mu_s L_e \int_{\theta_{min}}^{\theta_{max}} e^{-\mu_t (D \tan \theta + \Delta + \frac{D}{\cos \theta})} (A \cos \theta + B \sin \theta) d\theta. \quad (4.46)$$

Unfortunately this integral does not seem to have an analytical solution, even if we exclude the transmittance from the light, thus we attempt to approximate it.

4.3.2 Semi-Analytical Solution

Now that we have demonstrated how integrating the product of the geometry term and the transmittance terms does not lead to an analytical solution, we attempt to get as close as possible to the solution by approximating only the transmittance terms and not the full integrand.

Taylor approximation

We choose to approximate the product of the transmittance terms using a Taylor expansion:

$$\begin{aligned} T_{CL}(\theta) &= T_C(\theta) T_L(\theta) = e^{-\mu_t (D \tan \theta + \Delta + \frac{D}{\cos \theta})} \\ \tilde{T}(\theta) &= \sum_{n=0}^K \frac{T_{CL}^{(n)}(a)}{n!} (\theta - a)^n. \end{aligned} \quad (4.47)$$

Because of the polynomial nature of Taylor expansions, we are able to solve the approximated integral of equation 4.46 analytically and as a result obtain an importance sampling scheme that is approximately proportional to the product of the geometry term and the transmittance terms,

$$L(\mathbf{x}, \mathbf{z}) \approx \mu_s L_e \int_{\theta_{min}}^{\theta_{max}} \tilde{T}(\theta) (A \cos \theta + B \sin \theta) d\theta. \quad (4.48)$$

It is well known that exponential functions are difficult to fit as many terms are normally needed for good accuracy as such there are not many choices of approximations that will work besides a Taylor expansion since an exponential function can be expressed perfectly as a Taylor series. We are able to use only a few terms since the transmittance function is a very smooth exponential decay. Previous work such as Pegoraro et Al.[PP09] also used Taylor expansions as approximation in the context of participating media. In practice using an order of $K = 6$ seems to provide the best trade-off between quality and computation time. The approximation is very good between the integration bounds θ_{min} and θ_{max} for common μ_t values. The resulting polynomial expression is:

$$\tilde{T}(\theta) = T_0 + T_1\theta + T_2\theta^2 + T_3\theta^3 + T_4\theta^4 + T_5\theta^5 + T_6\theta^6, \quad (4.49)$$

where T_n is the n^{th} coefficient of the Taylor expansion of $\tilde{T}(\theta)$. We provide the value of those coefficients in the appendices. We use the expansion point $a = 0$ as it is the middle value between $-\pi/2$ and $\pi/2$, resulting in the best approximation possible on average. We have also considered using the middle value between θ_{min} and θ_{max} as an expansion point, but we ended up not using it since the improved accuracy provided by the resulting expression is not sufficient to compensate for the additional computation time. We have also investigated approximating each transmittance term independently but interestingly the approximation seems to be better for the product of the terms. On figure 4.8 we can see in blue the transmittance function and in red its Taylor approximation.

PDF normalization

The angular PDF is computed as:

$$p(\theta) = c \tilde{T}(\theta) (A \cos \theta + B \sin \theta). \quad (4.50)$$

The normalization factor c is computed as:

$$\begin{aligned} c &= \frac{1}{\int_{\theta_{min}}^{\theta_{max}} \tilde{T}(\theta) (A \cos \theta + B \sin \theta) d\theta} \\ &= \frac{1}{Q(\theta_{max}) \cos \theta_{max} - Q(\theta_{min}) \cos \theta_{min} + P(\theta_{max}) \sin \theta_{max} - P(\theta_{min}) \sin \theta_{min}} \end{aligned} \quad (4.51)$$

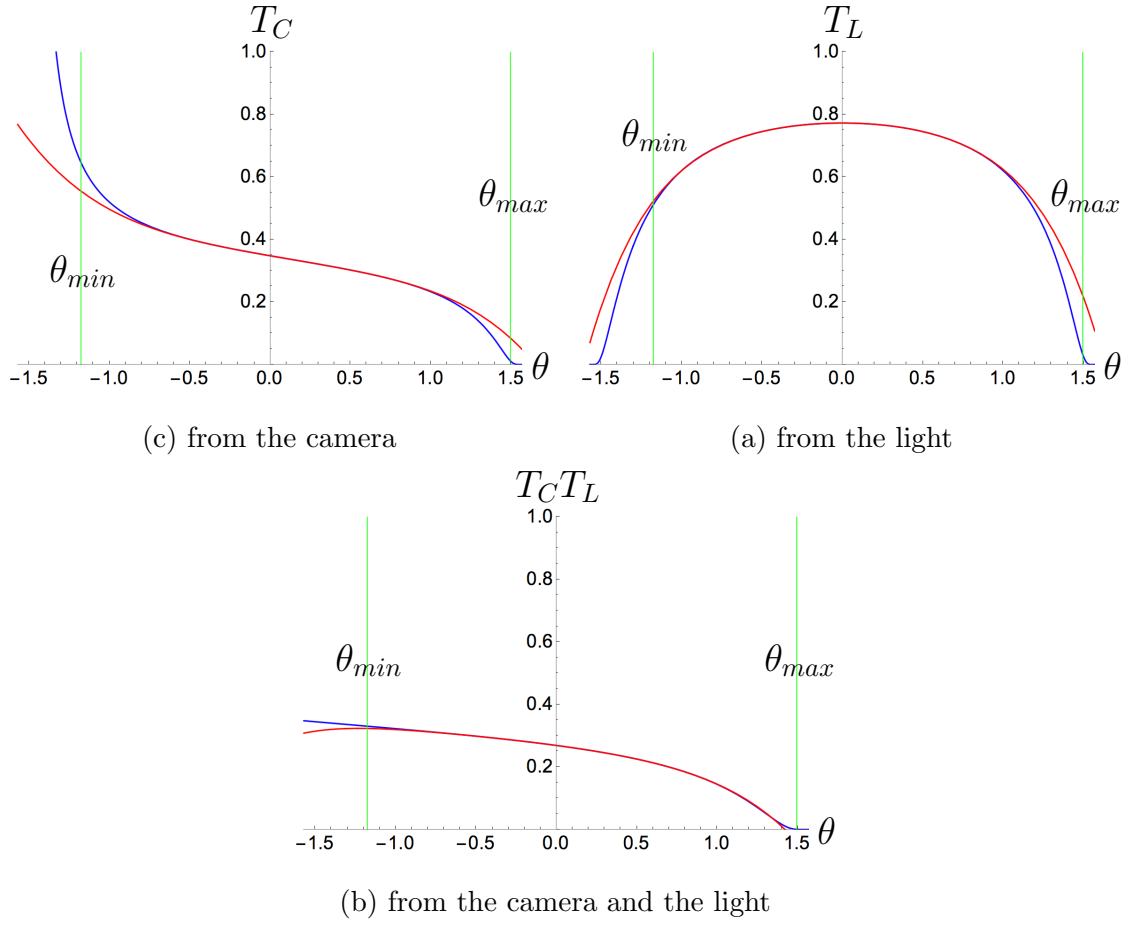


Fig. 4.8: Taylor approximation of transmittance.

$$Q(\theta) = \sum_{n=0}^6 Q_n \theta^n \quad P(\theta) = \sum_{n=0}^6 P_n \theta^n, \quad (4.52)$$

where Q_n is the n^{th} coefficient of the polynomial Q . We provide the value of those coefficients in the appendices. The full normalized PDF and CDF expressions are then:

$$p(\theta) = \frac{\tilde{T}(\theta) (A \cos \theta + B \sin \theta)}{Q(\theta_{max}) \cos \theta_{max} - Q(\theta_{min}) \cos \theta_{min} + P(\theta_{max}) \sin \theta_{max} - P(\theta_{min}) \sin \theta_{min}} \quad (4.53)$$

$$P(\theta) = \frac{Q(\theta) \cos \theta - Q(\theta_{min}) \cos \theta_{min} + P(\theta) \sin \theta - P(\theta_{min}) \sin \theta_{min}}{Q(\theta_{max}) \cos \theta_{max} - Q(\theta_{min}) \cos \theta_{min} + P(\theta_{max}) \sin \theta_{max} - P(\theta_{min}) \sin \theta_{min}}. \quad (4.54)$$

Note that in rare cases such as very large μ_t value, the approximation can be bad and the positivity constraint of a PDF can be violated. We thus clamp the PDF and CDF value to zero to ensure that the PDF always stays positive in those cases.

Sampling

Since the CDF is an expression involving two polynomials of order 6, it does not have a known analytical invert. As such we use the *Newton-Raphson method* to numerically invert the CDF. This method has a quadratic convergence rate, meaning that the number of correct digits of precision doubles at every iteration. While this method is simple and fast, it necessitates to compute the derivative of the function to invert. Luckily in our case the derivative of the CDF is equivalent to the PDF. The initial guess is set to the middle value between θ_{min} and θ_{max} , bracketing the only possible root of the function in that range. Since a CDF is a monotonic non-decreasing function and is remarkably smooth within these bounds, very few iterations are needed to converge to a solution with an acceptable degree of precision. On average only about three iterations are needed and in the very rare case the absolute maximum seems to be about ten iterations.

4.4 Phase Function

The last term to importance sample is the phase function, we show how to extend our sampling scheme that importance sample the geometry term to also include the phase function in a similar manner than with the transmittance terms. As we are unable to find an analytical solution to compute the CDF of the product of the geometry term and the phase function, we approximate the phase function using a Taylor expansion, leading to a CDF of the same form as the CDF to importance sample the geometry term and the transmittance terms.

4.4.1 Analytical Solution

The illumination of a single point-normal ignoring transmittance is given by:

$$L(\mathbf{x}, \mathbf{z}) = \mu_s L_e \int_0^z f_p(\mathbf{y}, \mathbf{x}, \mathbf{u}) \frac{|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|}{\|\mathbf{y} - \mathbf{u}\|^2} dy. \quad (4.55)$$

Isotropic

In the case of isotropic phase function,

$$f_p(\mathbf{y}, \mathbf{x}, \mathbf{u}) = \frac{1}{4\pi}, \quad (4.56)$$

it does not require any special treatment since it is a constant and can be easily extracted out from the CDF integral.

Anisotropic

It is however a lot more challenging for an anisotropic phase function like the Henyey-Greenstein (HG) model:

$$f_p(\mathbf{y}, \mathbf{x}, \mathbf{u}) = f_p(\alpha) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g \cos \alpha)^{3/2}}, \quad (4.57)$$

where $\cos \alpha = \boldsymbol{\omega} \cdot -\bar{\boldsymbol{\omega}}$ since the scattering of light is not the same everywhere around the shading point. Observing that the HG phase function is already parametrized in the angular domain, we only need to sample an angle proportionally to the product of the phase function and the geometry term. Once this angle is sampled, it can be projected back to a distance on the ray in the same way as before. A small issue is that the angle α is referenced from the ray's direction $\boldsymbol{\omega}$, but it can be mapped to the equi-angular frame by simply offsetting it by $+\pi/2$ like seen on figure 4.9.

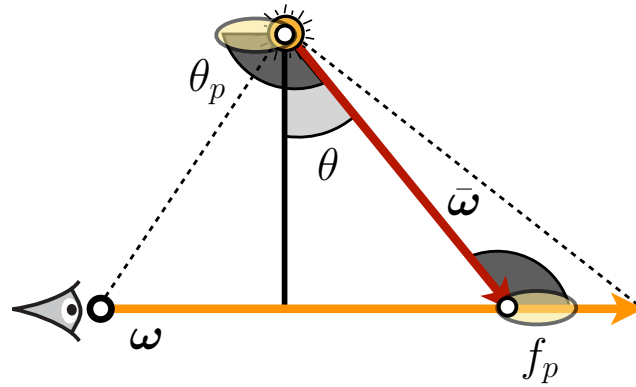


Fig. 4.9: Anisotropic phase function in equi-angular frame.

The resulting expression of the phase function is then:

$$f_p(\theta) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 + 2g \sin \theta)^{3/2}}. \quad (4.58)$$

The phase function is a spherical density distribution of scattered radiance, as such it is defined over the spherical domain S^2 :

$$\int_{S^2} f_p(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) d\bar{\boldsymbol{\omega}} = \int_0^\phi \int_0^\alpha f_p(\alpha) \sin \alpha d\alpha d\phi = \int_0^\phi \frac{1}{2\pi} d\phi \int_0^\alpha 2\pi f_p(\alpha) \sin \alpha d\alpha. \quad (4.59)$$

Since the angle ϕ does not change along the ray and the phase function is symmetrical around the ray we can ignore this angle. By ignore the angle ϕ and adding in the expression of the geometry term derived previously we get:

$$\begin{aligned} L(\mathbf{x}, \mathbf{z}) &= \mu_s L_e \int_{\theta_{min}}^{\theta_{max}} 2\pi f_p(\theta) \sin(\theta + \pi/2) (A \cos \theta + B \sin \theta) d\theta \\ &= \mu_s L_e \frac{1 - g^2}{2} \int_{\theta_{min}}^{\theta_{max}} \frac{(A \cos \theta + B \sin \theta) \cos \theta}{(1 + g^2 + 2g \sin \theta)^{3/2}} d\theta. \end{aligned} \quad (4.60)$$

By doing a judicious change of variable $u = \sin \theta$ we obtain the simplified expression:

$$L(\mathbf{x}, \mathbf{z}) = \mu_s L_e \frac{1 - g^2}{2} \int_{u_{min}}^{u_{max}} \frac{A\sqrt{1 - u^2} + Bu}{\sqrt{1 + g^2 + 2gu}^3} du. \quad (4.61)$$

Unfortunately no analytical solution for this integral seems to exist, thus we approximate it using a Taylor expansion very similarly to what we did for the transmittance terms.

4.4.2 Semi-Analytical Solution

Since we are unable to find an analytical solution we use the same method that we have used to importance sample the transmittance terms. We approximate the phase function as a Taylor expansion:

$$\tilde{F}(\theta) = 2\pi f_p(\theta) \cos(\theta) = \sum_{n=0}^K \frac{f_p^{(n)}(a)}{n!} (\theta - a)^n. \quad (4.62)$$

Resulting in the following approximation of the illumination:

$$L(\mathbf{x}, \mathbf{z}) \approx \mu_s L_e \int_{\theta_{min}}^{\theta_{max}} \tilde{F}(\theta) (A \cos \theta + B \sin \theta) d\theta. \quad (4.63)$$

A Taylor expansion of order $K = 6$ can be used, but since the phase function is very smooth we only need to use a 4th order polynomial (See figure 4.10). We also use an expansion point $a = 0$ for the same reasons that we have explained in the previous section. Previous work such as Pegoraro et Al.[PP09] also propose to use Taylor expansions as approximation to anisotropic phase function such as the HG model. The resulting PDF and CDF expression has the same form than the transmittance one, only the Taylor coefficients change:

$$p(\theta) = \frac{\tilde{F}(\theta) (A \cos \theta + B \sin \theta)}{Q(\theta_{max}) \cos \theta_{max} - Q(\theta_{min}) \cos \theta_{min} + P(\theta_{max}) \sin \theta_{max} - P(\theta_{min}) \sin \theta_{min}} \quad (4.64)$$

$$P(\theta) = \frac{Q(\theta) \cos \theta - Q(\theta_{min}) \cos \theta_{min} + P(\theta) \sin \theta - P(\theta_{min}) \sin \theta_{min}}{Q(\theta_{max}) \cos \theta_{max} - Q(\theta_{min}) \cos \theta_{min} + P(\theta_{max}) \sin \theta_{max} - P(\theta_{min}) \sin \theta_{min}}. \quad (4.65)$$

We provide the value of those coefficients in the appendices. The same technique can be used to importance sample the product of the geometry, transmittance and phase function terms, but the resulting CDF expression becomes quite expensive to evaluate, as such an MIS scheme seems to be a better alternative. It is also possible to use a Taylor expansion to approximate the product of the transmittance and phase function together instead of separately, but fitting the resulting function seems to be challenging due to the different nature of the two terms.

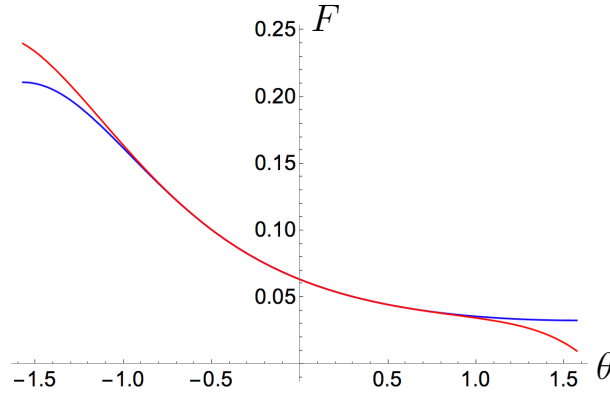


Fig. 4.10: Taylor approximation of the HG phase function.

4.5 Importance Sampling Polygonal Lights

We have presented in the previous sections our sampling scheme for a single point-normal in a medium, by importance sampling the geometry term, the transmittance terms and the phase function. We now show how to apply this scheme to importance sample a polygonal light efficiently by sampling a distance using a simple tabulation scheme of a small set of PNs at the surface of the polygonal light and by sampling a direction using a variant of the subtended solid angle sampling method. Going back to the original problem of rendering polygonal lights in participating media:

$$\begin{aligned}
 L(\mathbf{x}, \mathbf{z}) &= \int_0^z T(\mathbf{x}, \mathbf{y}) \mu_s \int_{A_e} f_p(\mathbf{y}, \mathbf{x}, \mathbf{u}) T(\mathbf{y}, \mathbf{u}) L_e(\mathbf{y}, \mathbf{u}) G(\mathbf{y}, \mathbf{u}) dA(\mathbf{u}) dy \\
 &\approx \frac{1}{N} \sum_{i=1}^N \frac{T(\mathbf{x}, \mathbf{y}) \mu_s f_p(\mathbf{y}, \mathbf{x}, \mathbf{u}) T(\mathbf{y}, \mathbf{u}) L_e(\mathbf{y}, \mathbf{u}) G(\mathbf{y}, \mathbf{u})}{p(y)p(\bar{\omega})}.
 \end{aligned} \tag{4.66}$$

We have demonstrated previously how the illumination of a polygonal light can be accurately approximated using a small set of PNs at the surface of the light:

$$L(\mathbf{x}, \mathbf{z}) \approx \mu_s L_e \sum_{k=1}^M \int_0^z T(\mathbf{x}, \mathbf{y}) f_p(\mathbf{y}, \mathbf{x}, \mathbf{u}_k) T(\mathbf{y}, \mathbf{u}_k) G(\mathbf{y}, \mathbf{u}_k) dy. \tag{4.67}$$

As $M \rightarrow \infty$ the contribution of all PNs converges to the contribution of the polygonal light. In practice however M does not have to approach infinity to get a good approximation. In the case of a rectangular light, we can get pretty close using as few as 12 PNs. To render a polygonal light we thus randomly choose one PN out of the set of M PNs and sample the distance into the medium by importance sampling the product of the geometry, transmittance and phase function terms of the chosen PN. The problem of rendering a polygonal light in participating media becomes then a problem of importance sampling M PNs, the only missing part is to design an efficient way to choose one PN.

The naive way to accomplish this would be to choose a PN uniformly, but the best way is to choose the PN according to its contribution relative to the total contribution of the M PNs. If the number of PNs is small, it is possible to create a discrete *tabulated CDF* that weights the probability of sampling each PN according to its contribution, making it more likely to sample the PNs that have a bigger impact on the illumination. This tabulated CDF

depends on the camera ray, thus it needs to be recomputed at each pixel sample. While this is very accurate it can become quite expensive as the number of PNs grows. As an attempt to mitigate this problem, we have investigated using an MIS scheme instead of a tabulation scheme, but the variance reduction seems to be worst for essentially the same computation time.

Until now we have only worried about the distance sampling scheme, yet once a distance into the medium has been sampled, a direction around that point also needs to be sampled. A naive way to accomplish this would be to sample the direction uniformly according to the area of the light source. A much better way is to sample the direction proportionally to the subtended solid angle of the light source, however the scheme used to sample the direction should work conjointly with the distance sampling scheme to efficiently reduce the variance of the Monte Carlo estimator. Thus we propose a variant of the subtended solid angle scheme where the polygonal light is seen as subdivided into M smaller polygonal lights on which a PN is placed at its center (See figure 4.11). A random direction $\bar{\omega}$ is then sampled proportionally to the subtended solid angle of the sub-polygon $\Omega_e^k(y)$ associated to the chosen PN. In the case where a subtended solid angle scheme is not available for a given polygonal light, the random direction can be sampled uniformly according to the sub-polygon's area.

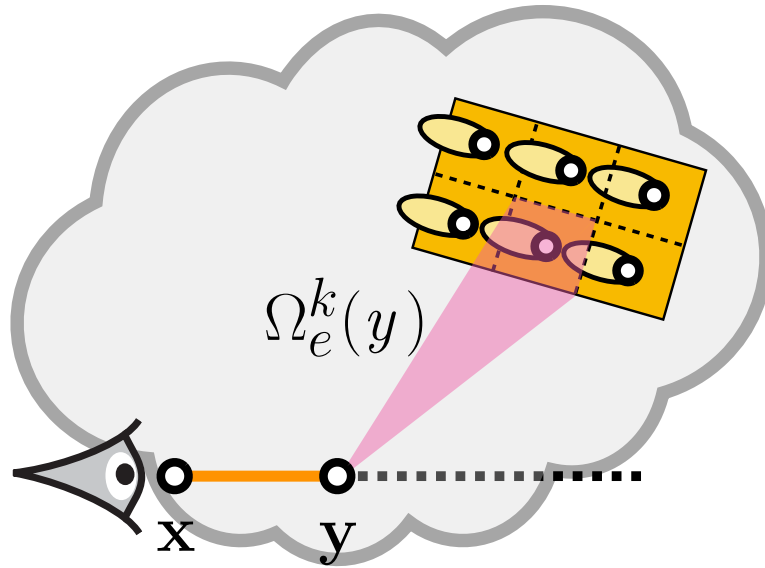


Fig. 4.11: Subtended solid angle of a sub-polygon.

4.6 Importance Sampling Mesh Lights

Since mesh lights can have arbitrary shapes, it is naturally much more challenging to render scenes using them rather than using simple well defined shapes like polygonal lights. Regardless of how complex their shape might be, mesh lights can be modeled as a finite set of triangle lights, approximating their true smooth surface. In this model, the tabulation scheme that we have developed for simple polygonal lights can also be applied to mesh lights since they can be seen as triangle lights which are really polygonal lights. Contrary to polygonal lights, the normal's direction of the PNs at the surface of a mesh light can be different, thus the contribution along its surface can vary greatly. As a result, a much greater number of PNs is necessary to approximate accurately the illumination of a mesh light. By simply placing a point-normal at the center of each of those triangle lights and creating a tabulated CDF to sample a distance according to the contribution of all the PNs works as expected but it becomes too expensive to be usable. To compute the tabulated CDF all the PNs need to be evaluated for each distance sample. It is easy to see that this method does not scale since the computation time required to sample a distance increases linearly with the number of PNs used to approximate the mesh light. In other words, the more complex the mesh light becomes, the more triangles it will have thus the more expensive it will be to take a distance sample. Thus to be usable in a production renderer the solution needs to be scalable, meaning it should not become much more expensive to take a sample as the complexity of the mesh light's grows.

4.6.1 Lights Clustering

To solve this problem we want to offload as much computation as possible to a preprocessing stage that depends on the mesh light geometrical definition (the position and normal of each of its triangles) and minimize the computation happening at every sample during the shading stage. Previous works such as Lightcuts[WFA⁺05] already tackled this problem by using a hierarchical light clustering strategy that is sub-linear on the number of PNs. This work takes advantage of two observations. First, many PNs are quite similar on a mesh light and we can take advantage of their similarity to, in some sense, share parts of their computation since they will result in very similar contribution to the illumination. If some PNs are similar enough, we can essentially merge them into one *representative* PN and compute the illumination of the representative PN instead of computing the illumination of

each PN out of a potentially large set of similar PNs. We mean by similar how close they are in space and if they are pointing roughly in the same direction. While grouping PNs into clusters of PNs we also want to encode the spatial and directional similarity of the clusters using a hierarchical data structure like a tree to allow fast retrieval later on during shading. Secondly, in typical scenes most PNs do not contribute much to the illumination. In some cases they might not even contribute at all, for example if the point-normal is backfacing the camera ray. The subset of PNs that contribute substantially to the illumination depends on the camera ray, and so can potentially change at every pixel of the rendered image. It is possible to find this subset during shading and evaluate only this subset instead of the whole set for every sample. To reduce this search we take advantage of the previous hierarchical clustering of PNs and using an iterative algorithm to traverse the tree we determine the subset that contributes the most to the given camera ray.

4.6.2 Volumetric Lightcuts

Follow up work to Lightcuts[WABG06] applied it to participating media but relies on discretizing the distance integral of a medium into many gather-light pairs that necessitates additional hierarchical data structures. Those data structures are then used to partition the gather-light pairs into clusters based on a maximum error bound. The resulting algorithm diverges quite a bit from the original lightcuts for surfaces and is not a simple extension. In the previous section we have derived an analytical solution to the illumination of a single point-normal in a medium. The point-normal in a volume is essentially the analog of the oriented point light in the original Lightcuts for surfaces. Thus it appears natural to use the same algorithm as Lightcuts but using a different entity for which we can compute the full illumination along a given ray using our analytical solution of the geometry term of a point-normal light. It is even possible to take into account the transmittance and the phase function using the Taylor approximations that we have used previously. To our knowledge this is the first implementation of a volumetric lightcuts done in this manner. This technique is normally implemented in two stages: preprocessing stage and shading stage. The preprocessing stage happens once before any shading is done and its main job is to build the hierarchical data structure that is going to be traversed to evaluate the variable set of PNs during the shading stage each time a pixel needs to be shaded.

Preprocessing Stage

The first step, before doing any shading is a preprocessing step where we build a binary tree commonly called a *light tree* (See figure 4.12) to cluster the lights in a very similar manner than Lightcuts. Each node of this tree represents a cluster of PNs and the leafs are single PNs. One of the child of the node is chosen randomly as a representative to minimize the computation of the illumination. A cluster is defined by an axis-aligned bounding box, a bounding cone of directions and its total intensity. The total intensity I_c of a cluster is simply the sum of the intensity of both of its children. The light tree is built using a bottom-up greedy algorithm. Starting from the whole set of PNs, we take the closest pair of PNs and cluster them together. To cluster the lights we use a similarity metric based on position and direction:

$$I_c(\alpha_c^2 + c^2(1 - \cos \beta_c)^2). \quad (4.68)$$

where α_c is the diagonal length of the cluster bounding box and β_c is the half-angle of its bounding cone. The coefficient c defines how important the direction is relatives to the position. We set this constant to the diagonal of the scene's bounding box like suggested in the Lightcuts paper. This algorithm in its most naive form has a runtime complexity of $O(N^3)$, but the Lightcuts work proposes a faster way to build the tree that is $O(N^2)$ using [WFA⁺05] and an optimized implementation in Walter et al.[WBKP08].

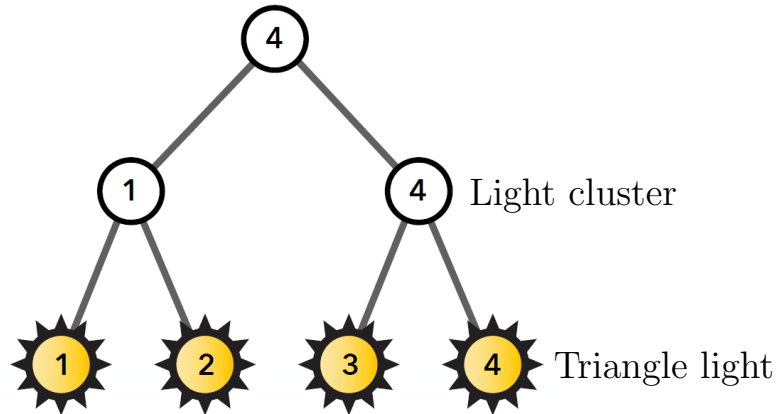


Fig. 4.12: Lightcuts data structure. Image source: [Now18]

Shading Stage

Once the light tree has been built, we traverse it to create a *lightcut*. A lightcut is essentially a specific clustering of lights that depends on the camera ray. It defines the resulting approximation of the illumination by considering only a specific set of PNs (See figure 4.13). A lightcut is created using an iterative algorithm that refines the cut until its approximation error is low enough. We use a perceptual error of 2% to define the minimum acceptable error. Starting from a lightcut containing only the root node of the light tree, the illumination and the error of the current cut is computed. If the error is too high, the cut is refined by adding the two children of the node with the highest error bound and removing that node from the cut. The illumination and error of both children are then computed and the total illumination of the lightcut is updated accordingly.

The computation of the illumination and error of a cut is where our technique differs from Lightcuts. We have seen previously that the illumination of a point-normal in a medium (ignoring transmittance and phase function) is given by:

$$L(\mathbf{x}, \mathbf{z}) = \mu_s L_e \int_0^z \frac{|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|}{\|\mathbf{y} - \mathbf{u}\|^2} dy. \quad (4.69)$$

We thus compute the illumination from a cluster of PNs (a single node in the light tree) using our analytical solution to this equation. We can also include the transmittance and phase function terms using the Taylor expansions developed previously but the illumination would then be biased. The maximum error bound is the metric that gives us an insight on how good the set of PNs from the current lightcut approximates the true illumination of the given ray from the whole set of PNs of the mesh light. It is the maximum possible illumination of a given cluster of PNs as such, to compute it we need to maximize the integral of equation 4.69. We propose to maximize each part of the geometry term $\frac{|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|}{\|\mathbf{y} - \mathbf{u}\|^2}$ of a PN separately. We first attempt to minimize the $1/\|\mathbf{y} - \mathbf{u}\|^2$ term by finding the closest point to the camera ray in the bounding box of a given cluster. We then maximize the $|n(\mathbf{u}) \cdot (\mathbf{y} - \mathbf{u})|$ term by finding the direction in the bounding cone that produces a dot product closest to one, its maximum possible value. Finally we compute the illumination of the resulting PN using our analytical solution of equation 4.69. It should be noted that maximizing the error this way does not guarantee the global maximum since we really want to maximize the integral of both terms not of each term independently. To reduce the number of PNs in a cut, we need

to compute a conservative error bound that is as tight as possible. Note that this is still subject of ongoing research and the details to compute the maximum error bound efficiently are still uncertain.

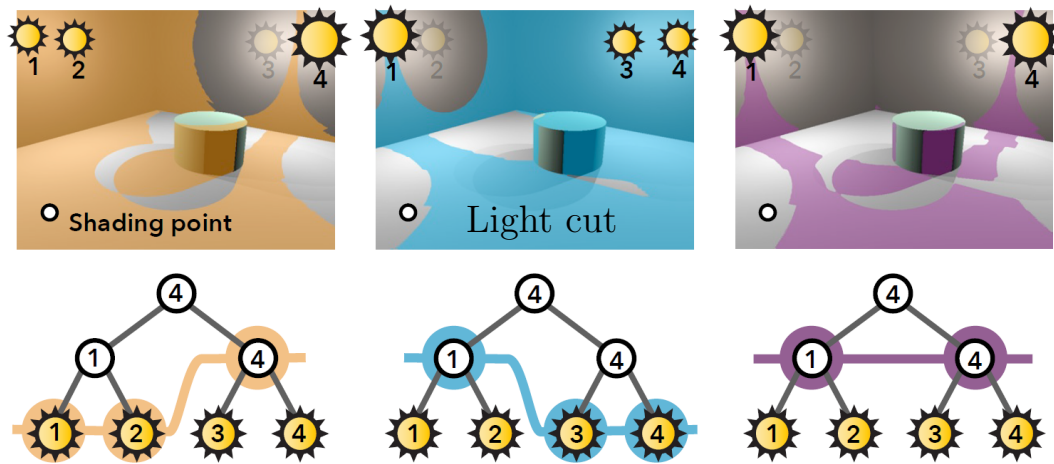


Fig. 4.13: Lightcuts. *Image source:* [Now18]

CHAPTER 5

Results

We now present the results of our technique on two types of lights: a point-normal light and a rectangular light. We compare our method to equi-angular sampling[KF12] which is currently the state of the art for polygonal lights importance sampling in participating media. We then analyze the convergence rate of our technique and finally we provide an analysis of the Newton solver used to invert the CDF to importance sample the transmittance term.

5.1 Implementation

We have implemented our technique in a path tracer similar to the one that is used today in the film and visual effects industry. Our renderer is a highly modified version of *Nori* which is itself a fork of *Mitsuba* [Jak10], a well known renderer that is considered the reference in rendering research. We have compared our references images to the ones rendered using *Mistuba* to validate our renderer. Our volumetric path tracer only handles homogeneous participating media but our technique can also be used in a full-featured renderer with heterogeneous participating media and non-zero emission or absorption. We have only tested our polygonal importance sampling strategy on a rectangular light but it can be easily applied to other shapes as well. We use the method proposed in Urena et al.[UFK13] to sample a direction proportionally to the subtended solid angle of a rectangular light or a sub-rectangle of this light.

5.2 Technique Comparison

We compare our technique to equi-angular sampling on two scenes composed only of a single light in an infinite homogeneous medium. The only difference between those two scenes is the type of light used. These scenes are deliberately kept simple to limit the sources of noise. We choose to first show results with transmittance artificially disabled in order to make sure that the transmittance term cannot be a source of noise, allowing us to better visualize the noise caused by the geometry term of a point-normal. For each result we also compare our technique to a converged image that has been rendered using the same path tracer without doing any importance sampling i.e. by uniform sampling.

5.2.1 Point-normal Light

The images from figure 5.1 - 5.4 compares our technique to equi-angular sampling for a scene with a single point-normal light in an infinite homogeneous medium.

Transmittance disabled

By artificially disabling the transmittance, we can see on figure 5.1c that our technique produces a noise-free image using only one sample. We show in 5.1a the original equi-angular sampling that does not take into account the clamped cosine of the point-normal, generating more noise than the clamped equi-angular sampling technique shown on figure 5.1b. While our technique has the most expensive sample cost it is not an issue since it converges in a single sample, resulting in an improvement of more than 3800% over clamped equi-angular sampling.

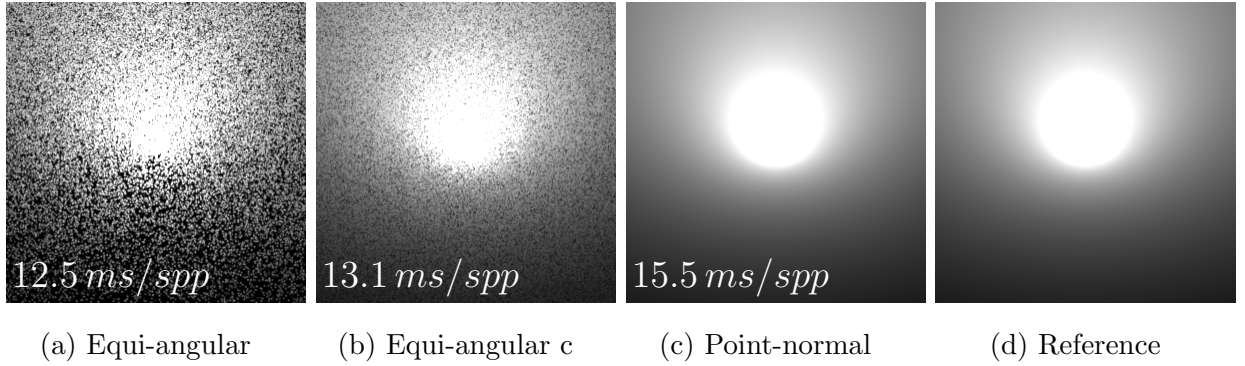


Fig. 5.1: One sample comparison of our technique on the point-normal light scene.

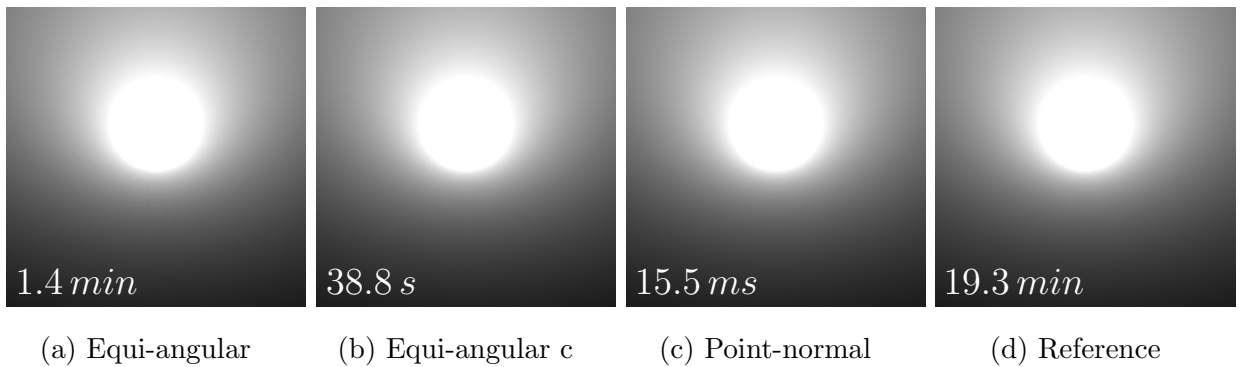


Fig. 5.2: Converged images of every technique on the point-normal light scene.

Transmittance enabled

By re-enabling transmittance like it is normally rendered, we can see on figure 5.3b that the image produced by our technique that only importance sample the geometry term of the point-normal has a bit less noise than clamped equi-angular (figure 5.3a) but is still very noisy. Our technique that also importance sample the transmittance terms (figure 5.3c) leads to an almost noise-free result. The few grains of noise on figure 5.3c are only caused by our approximation of the transmittance terms. The considerable difference in sampling cost from both of our techniques is due to the Newton solver used in our semi-analytical solution. Nevertheless, our method is still about two times faster than equi-angular sampling on this scene. Note that even though the images here appears much darker than the images with transmittance disabled, it is still the same scene as before. The scene only appears darker because the light is attenuated by the medium according to the transmittance term.

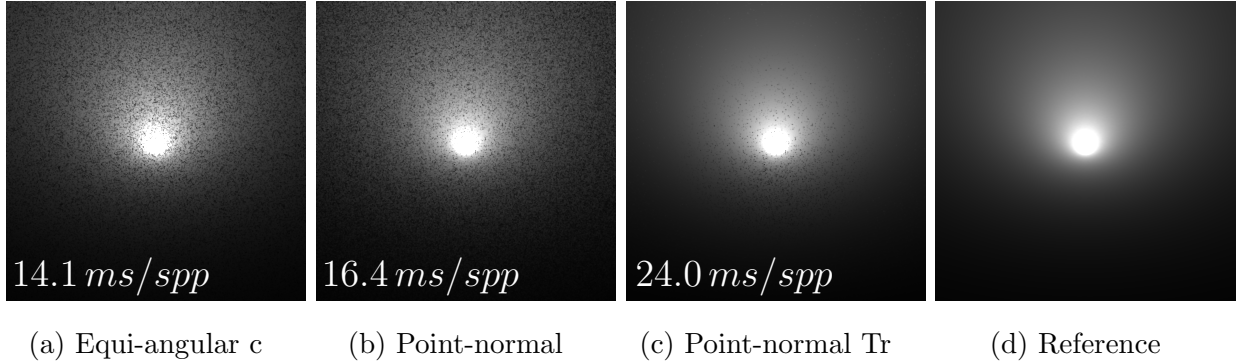


Fig. 5.3: One sample comparison of our technique on the point-normal light scene.

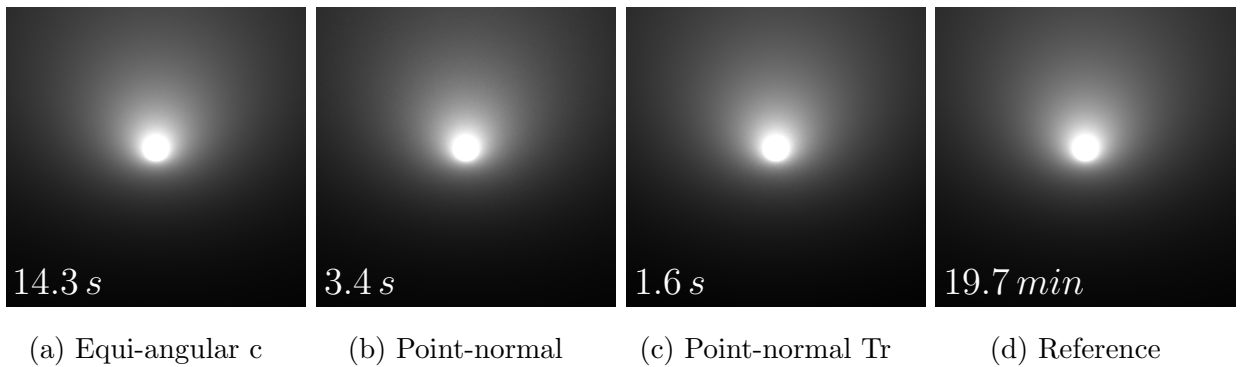


Fig. 5.4: Converged images of every technique on the point-normal light scene.

5.2.2 Rectangular Light

The images from figure 5.5 - 5.8 compares our technique to equi-angular sampling for a scene with a single rectangular light in an infinite homogeneous medium.

Transmittance disabled

We again first show results by disabling transmittance to isolate the sources of noise for better visualization. On figure 5.5b we show our technique that importance sample the geometry term of only a single point-normal placed at the center of the rectangular light. Figure 5.5c shows our polygonal importance sampling scheme that samples a distance using a tabulation scheme of 12 PNs. The direction is sampled proportionally to the subtended solid angle of the sub-rectangle associated to the chosen PN like explained in *Chapter 4*.

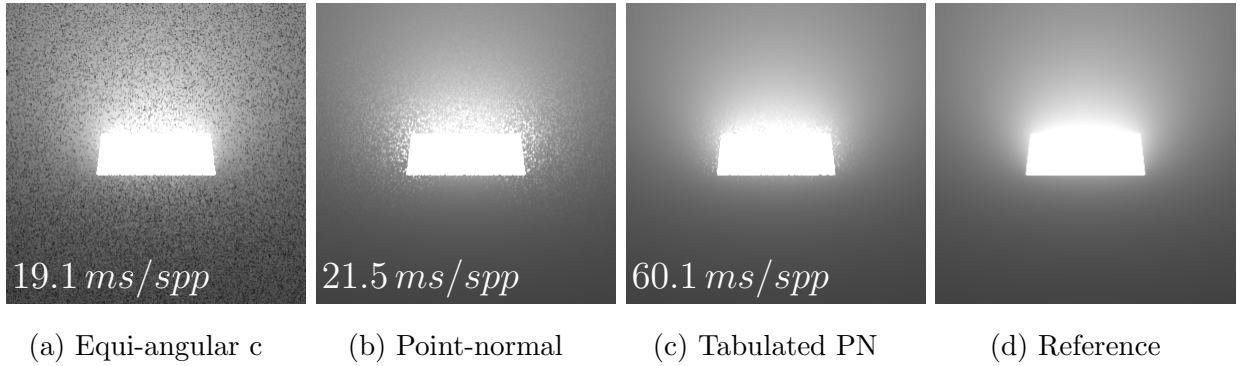


Fig. 5.5: One sample comparison of our technique on the rectangular light scene.

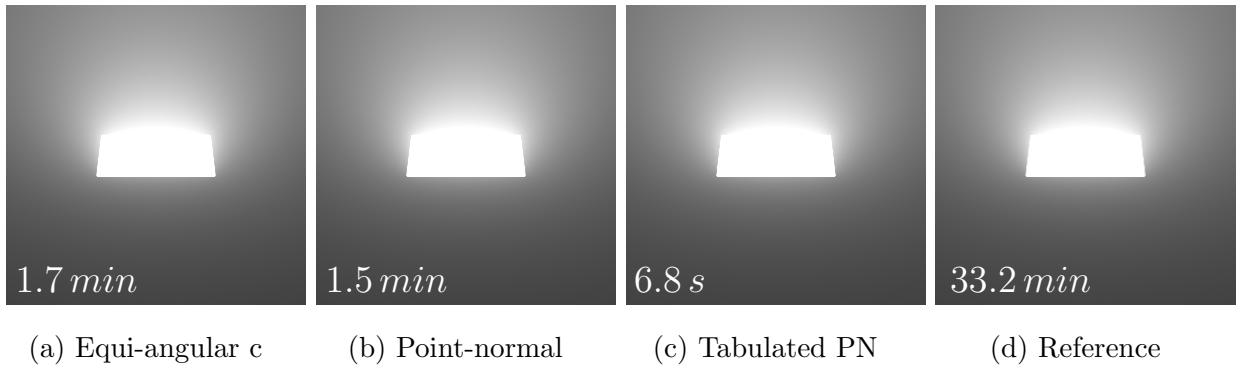


Fig. 5.6: Converged images of every technique on the rectangular light scene.

We can see that our technique from figure 5.5b massively reduces the noise compared to clamped equi-angular sampling (figure 5.5a). In addition our technique from figure 5.5c produces an image that is very close to noise free, the only source of noise is caused by the approximated illumination of the small set of PNs. Adding more PNs would of course reduce the noise but at the cost of increasing the cost per sample. Using 12 PNs seems to provide the best tradeoff. Our technique allows us to obtain a converged image about 15x faster than clamped equi-angular.

Transmittance enabled

We obtain similar results when the transmittance is enabled. On figure 5.7b and figure 5.7c our techniques are using the CDF that also importance sample the transmittance terms. While the gains are less impressive, we still get about 6.3x increase on converged rendering.

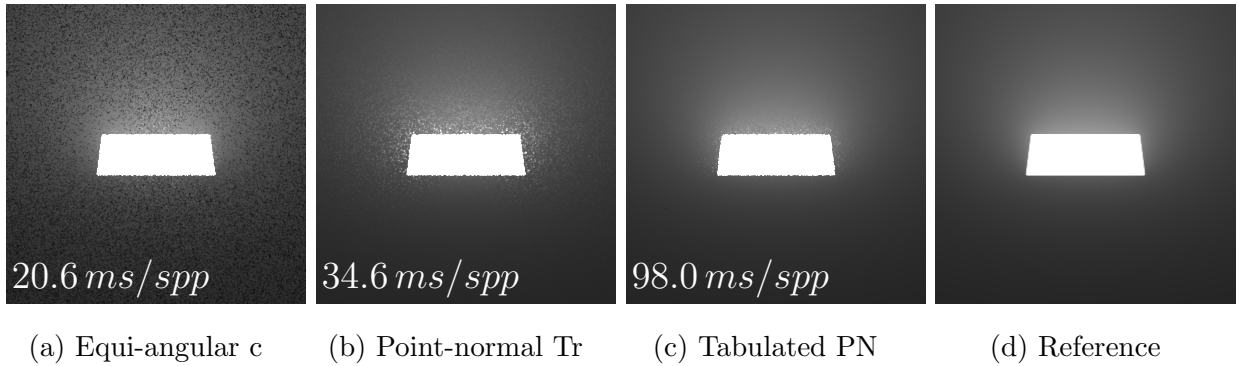


Fig. 5.7: One sample comparison of our technique on the rectangular light scene.

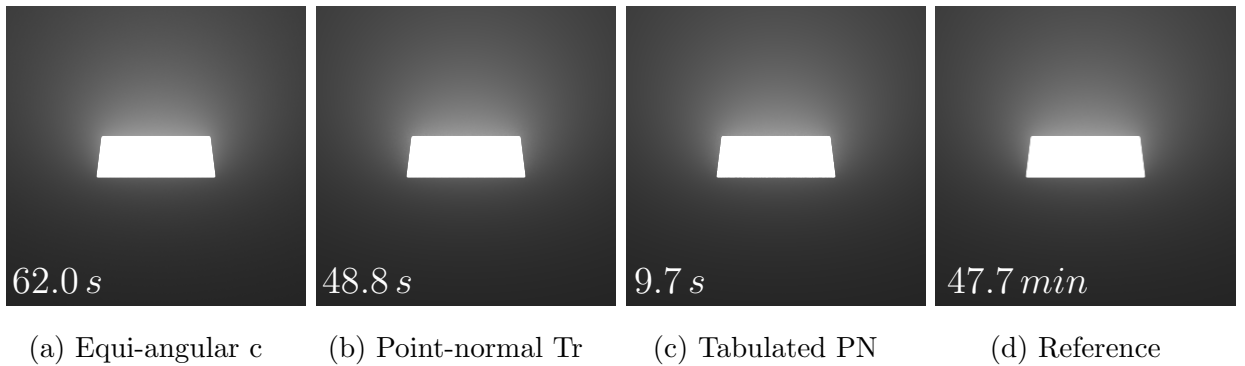


Fig. 5.8: Converged images of every technique on the rectangular light scene.

5.3 Convergence Analysis

We now show a plot on figure 5.9 to compare the convergence rate of our technique (in blue) to clamped equi-angular sampling (in green). The scene used is a single rectangular light in a very dense homogeneous medium ($\mu_s = 10.3$). The intensity of the light source is extremely high since very few amount of light can pass through this dense medium. We deliberately choose such a high μ_s value to get a very difficult scene to render, necessitating an enormous amount of samples to converge completely if not importance sampled properly. Our technique samples a distance using our tabulation scheme of 12 PNs and the direction is sampled proportionally to the subtended solid angle of the sub-rectangle associated to the chosen PN. The CDF used importance sample the product of the geometry and the transmittance terms. We see on figure 5.9 that from the start our technique has a RMSE more than two order magnitude lower than equi-angular. Since the blue curve is much lower than the green curve it is clear that our technique converges faster than equi-angular.

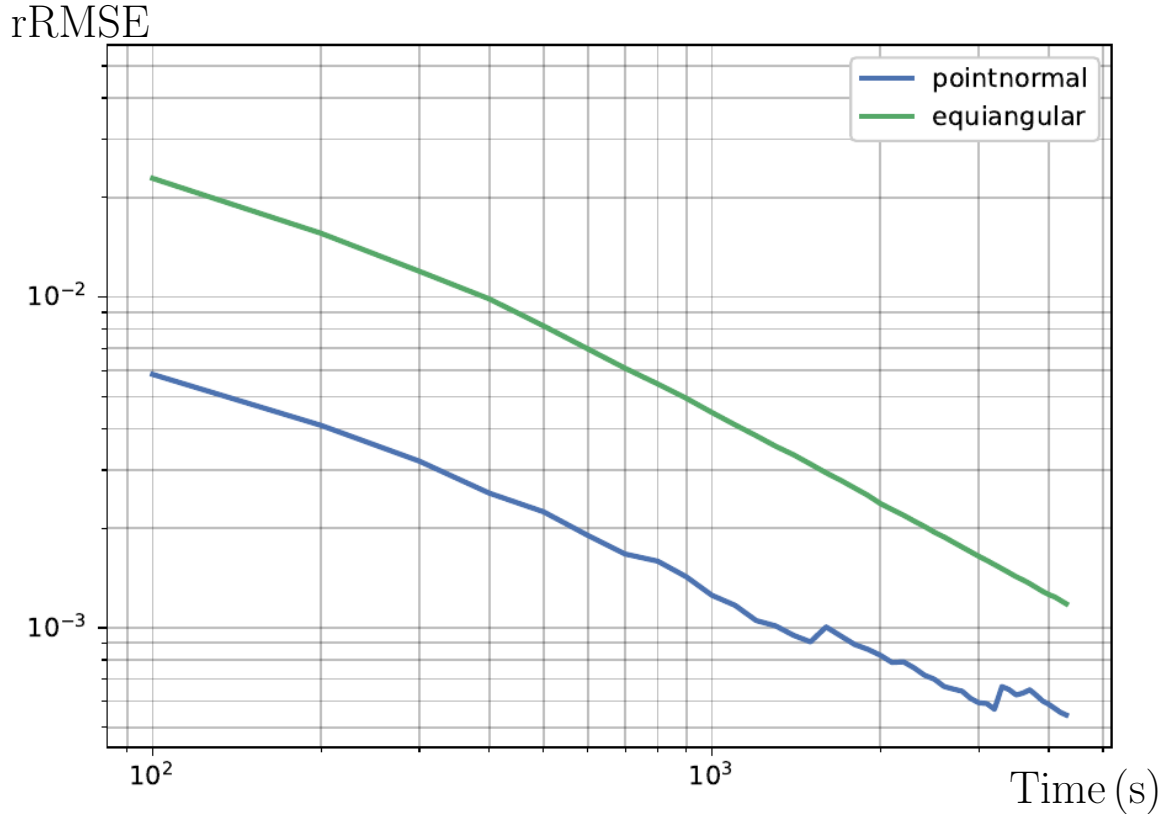


Fig. 5.9: Convergence rate of our technique compared to equi-angular sampling.

5.4 Newton Solver Analysis

To prove that our semi-analytical solution is usable even if it uses a Newton solver to invert the CDF, we show a visualization of the average number of iterations per pixel needed for a few scenes. On figure 5.11 we show that very few iterations are needed in practice. A value of one iteration is shown as dark blue and the maximum value of 10 iterations is shown as dark red. We can notice from this figure that the number of iterations is roughly the same on the whole image for both types of light sources except in the middle where the light source is. This result is expected since the CDF approaches a delta function when the distance between the camera ray and the light becomes small, which is more challenging to invert for a Newton solver. Another thing to observe is that the number of iterations almost never goes beyond 3 iterations, this confirms that the CDF is very smooth and thus easy to invert.

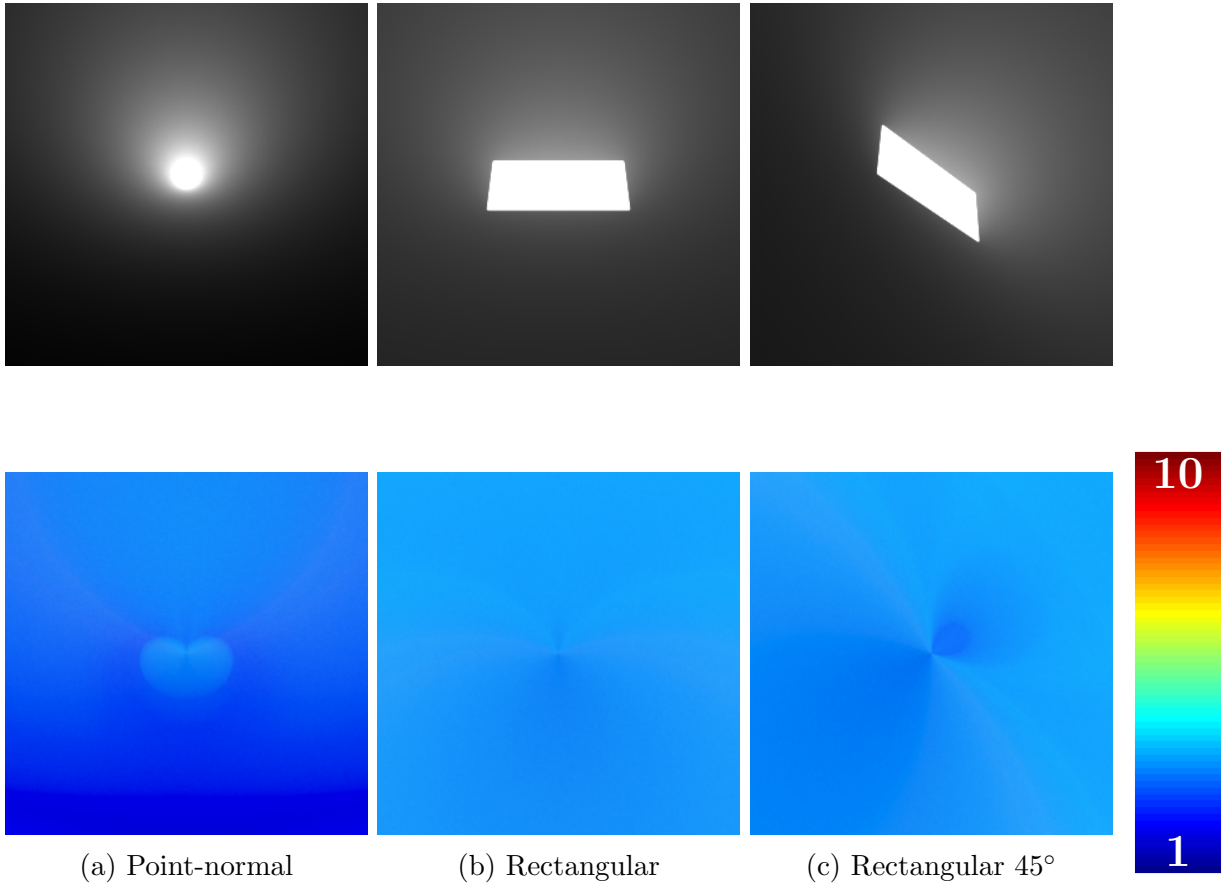


Fig. 5.11: Average iterations per pixel of the Newton-Raphson solver.

CHAPTER 6

Conclusion

Rendering in participating media is particularly challenging since the light is attenuated while it travels through the medium. We have seen that the illumination of scenes involving participating media can be computed using the *Volume Rendering Equation (VRE)*. This equation is normally solved via path tracing using various importance sampling schemes to reduce variance. The VRE involves many terms such as the geometry, the transmittance from the camera, the transmittance from the light source and the phase function. Ideally every term should be importance sampled if we want to get rid of the noise as fast as possible. Scenes with polygonal or mesh lights make the problem even more difficult since these lights have a surface, contrary to point lights. Every point along this surface emits light and need to be taken into account. An analytical solution to compute the illumination from light emitting surfaces in volumes is often impossible. We have shown that this problem is still hard even in the much simpler case of a point light with a direction (point-normal).

In the past, not much research tackled this problem as it seems fundamentally hard. We note the work of equi-angular sampling[KF12] that can be used by placing an imaginary point light at the center of a polygonal light and importance sample the geometry term of this imaginary light. We have demonstrated how this scheme is inefficient since the illumination of a polygonal light is concentrated in a particular direction normal to its surface, which differs greatly from the illumination of a point light where the light scatters from a single point in all directions equally.

We have first shown that importance sampling the full geometry term of a polygonal light does not seem to be possible since no analytical solution exists to compute the illumination from a polygonal light in a medium. We have thus demonstrated how a polygonal light can be seen as an infinite number of point-normal lights at its surface and how we can importance sample the geometry term of a single point-normal analytically using the same angular frame as equi-angular sampling. We have also derived a semi-analytical solution to importance sample the product of the geometry, the transmittance and the phase function terms of a point-normal. To take into account the transmittance and phase function terms we need to make an approximation using a Taylor expansion. We have shown that a 6th order polynomial is enough to approximate with good accuracy the transmittance term for common μ_s values. An anisotropic phase function can be approximated similarly using a 4th order Taylor expansion. We have then shown how to accelerate rendering of scenes with polygonal lights in participating media by sampling a distance proportionally to the contribution of a small set of PNs at its surface using a tabulation scheme. We have also proposed to sample a direction proportionally to the subtended solid angle of the sub-polygon associated to the chosen PN to make sure that the strategy used to sample a direction works conjointly with the one we have proposed to sample a distance. Our technique converges faster than equi-angular sampling and is directly applicable to production renderers.

6.1 Limitations

Our method is quite generic and as such can be applied to any type of polygonal lights however, like any other technique it has some limitations. To be used at its full efficiency a sampling scheme to sample a direction proportionally to the subtended solid angle of the polygonal light should be possible. In the failure case, sampling a direction uniformly to the area of the light source works decently. Furthermore, to handle transmittance and the phase function terms in the illumination of a point-normal we rely on a Taylor expansion of those terms, leading to only an approximation of the true solution. Unfortunately, since we are using an approximation in our importance sampling scheme, the resulting CDF is not guaranteed to be positive. This can become problematic in the case of big μ_s as it could produce biased results if not handled properly. In addition, the approximation gets worst as the μ_s value grows since the transmittance term approaches the delta function, which would theoretically require an infinite number of coefficients to be able to match the function.

While we have shown that our technique can be used to render mesh lights by using our light clustering strategy, in its current state it is not possible to use it in a path tracer such as the one used in the film and VFX industry. To do so would require to reformulate our technique into an importance sampling scheme.

6.2 Future Work

Our work constitutes a stepping stone for the challenging problem of rendering polygonal and mesh lights in participating media, but it is nowhere close to a full solution in the case of mesh lights and many things could be improved in the case of polygonal lights. To importance sample polygonal lights using PNs, we have proposed to subdivide the polygonal light and place a PN at the center of each sub-polygon. This uniform distribution of PNs on the polygonal light is not optimal, it might be better to use a non-uniform PN distribution scheme and place them at strategic locations to make a better use of the PNs budget. For example, in the case of a rectangular light it might be more advantageous to place more PNs at the edge of the light than in the middle since the edges really define the shape of the light and hence its contribution. Ideas from work using the *Finite Element Method* for example could be applied to this problem to improve the distribution of the PNs on the surface of the polygonal light.

Furthermore, we have shown that even attempting to analytically compute the illumination of a single point-normal is difficult and forced us to use approximations as some parts of the VRE like transmittance cannot be analytically integrated. We have said previously that we have failed to compute analytically the illumination of a polygonal light as this requires integrating the subtended solid angle of the light. Since we need to approximate the illumination in the end if we want to include the transmittance terms, it could be better to approximate the subtended solid angle of a polygonal light and integrate this approximation instead of the individual PNs.

The biggest remaining issue of our work is rendering of mesh lights efficiently. We have briefly shown how we can use our semi-analytical solution to compute the illumination of a single point-normal. A point-normal in a volume allows us to extend Lightcuts[WFA⁺05] to participating media in a more natural way than previous work, but we have used a relatively loose maximum error bound; resulting in the evaluation of more PNs than necessary. As such, it might be possible to develop a tighter error bound that would reduce the number

of PNs that needs to be evaluated on average. Finally, since Lightcuts[WFA⁺05] is not the technique currently used for light transport in industrial renderers, we need to find a way to use our hierarchical light clustering strategy in a path tracer. The most promising way seems to be to transform our light clustering strategy to an importance sampling scheme.

APPENDIX A

Appendices

A.1 Taylor expansion of the transmittance terms

$$\begin{aligned}
 T_{CL}(\theta) &= T_C(\theta)T_L(\theta) = e^{-\mu_t(D \tan \theta + \Delta + \frac{D}{\cos \theta})} \\
 \tilde{T}(\theta) &= \sum_{n=0}^6 \frac{T_{CL}^{(n)}(0)}{n!} \theta^n \\
 &= T_0 + T_1 \theta + T_2 \theta^2 + T_3 \theta^3 + T_4 \theta^4 + T_5 \theta^5 + T_6 \theta^6
 \end{aligned} \tag{A.1}$$

$$\begin{aligned}
 T_0 &= e^{-\mu_t(D+\Delta)} \\
 T_1 &= -u_1 \\
 T_2 &= \frac{1}{2}(T_1 + u_2) \\
 T_3 &= \frac{1}{6}(2T_1 + 3u_2 - u_3) \\
 T_4 &= \frac{1}{24}(5T_1 + 11u_2 - 6u_3 + u_4) \\
 T_5 &= \frac{1}{120}(16T_1 + 45u_2 - 35u_3 + 10u_4 - u_5) \\
 T_6 &= \frac{1}{720}(61T_1 + 211u_2 - 210u_3 + 85u_4 - 15u_5 + u_6) \\
 u_n &= \mu_t^n D^n T_0
 \end{aligned} \tag{A.2}$$

A.2 Polynomial coefficients of the transmittance CDF

$$c = \frac{1}{Q(\theta_{max}) \cos \theta_{max} - Q(\theta_{min}) \cos \theta_{min} + P(\theta_{max}) \sin \theta_{max} - P(\theta_{min}) \sin \theta_{min}} \quad (\text{A.3})$$

$$Q(\theta) = \sum_{n=0}^6 Q_n \theta^n \quad P(\theta) = \sum_{n=0}^6 P_n \theta^n \quad (\text{A.4})$$

$$\begin{aligned} Q_0 &= -B T_0 + 1A T_1 - 6A (T_3 - 20T_5) + 2B (T_2 - 12T_4 + 360T_6) \\ Q_1 &= -B T_1 + 2A T_2 + 6B (T_3 - 20T_5) - 24A (T_4 - 30T_6) \\ Q_2 &= -B T_2 + 3A T_3 - 60A T_5 + 12B (T_4 - 30T_6) \\ Q_3 &= -B T_3 + 4A T_4 + 20B T_5 - 120A T_6 \\ Q_4 &= -B T_4 + 5A T_5 + 30B T_6 \\ Q_5 &= -B T_5 + 6A T_6 \\ Q_6 &= -B T_6 \\ P_0 &= A T_0 + 1B T_1 - 6B (T_3 - 20T_5) - 2A (T_2 - 12T_4 + 360T_6) \\ P_1 &= A T_1 + 2B T_2 - 6A (T_3 - 20T_5) - 24B (T_4 - 30T_6) \\ P_2 &= A T_2 + 3B T_3 - 60B T_5 - 12A (T_4 - 30T_6) \\ P_3 &= A T_3 + 4B T_4 - 20A T_5 - 120B T_6 \\ P_4 &= A T_4 + 5B T_5 - 30A T_6 \\ P_5 &= A T_5 + 6B T_6 \\ P_6 &= A T_6 \end{aligned} \quad (\text{A.5})$$

A.3 Taylor expansion of the HG phase function

$$f_p(\theta) = 2\pi \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 + 2g \sin \theta)^{3/2}} \cos(\theta)$$

$$\tilde{F}(\theta) = \sum_{n=0}^6 \frac{f_p^{(n)}(0)}{n!} \theta^n \quad (\text{A.6})$$

$$= F_0 + F_1\theta + F_2\theta^2 + F_3\theta^3 + F_4\theta^4 + F_5\theta^5 + F_6\theta^6$$

$$F_0 = \frac{G}{H^{3/2}}$$

$$F_1 = -\frac{3Gg}{H^{5/2}}$$

$$F_2 = \frac{15Gg^2}{2H^{7/2}}$$

$$F_3 = -\frac{35Gg^3}{2H^{9/2}} + \frac{Gg}{2H^{5/2}}$$

$$F_4 = \frac{315Gg^4}{8H^{11/2}} - \frac{5Gg^2}{2H^{7/2}} \quad (\text{A.7})$$

$$F_5 = -\frac{693Gg^5}{8H^{13/2}} + \frac{35Gg^3}{4H^{9/2}} - \frac{Gg}{40H^{5/2}}$$

$$F_6 = \frac{3003Gg^6}{16H^{15/2}} - \frac{105Gg^4}{4H^{11/2}} + \frac{Gg^2}{3H^{7/2}}$$

$$G = \frac{1 - g^2}{4\pi}$$

$$H = 1 + g^2$$

A.4 Polynomial coefficients of the HG phase function CDF

Same as the *polynomial coefficients of the transmittance CDF*, replacing T_n by F_n .

References

- [Arv95] James Arvo. Applications of irradiance tensors to the simulation of non-lambertian phenomena. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 335–342. ACM, 1995.
- [Cha13] Subrahmanyan Chandrasekhar. *Radiative transfer*. Courier Corporation, 2013.
- [GKH⁺13] Iliyan Georgiev, Jaroslav Krivanek, Toshiya Hachisuka, Derek Nowrouzezahrai, and Wojciech Jarosz. Joint importance sampling of low-order volumetric scattering. *ACM Trans. Graph.*, 32(6):164–1, 2013.
- [HG41] Louis G Henyey and Jesse L Greenstein. Diffuse radiation in the galaxy. *The Astrophysical Journal*, 93:70–83, 1941.
- [Jak10] Wenzel Jakob. Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>.
- [Kaj86] James T Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.
- [KF12] Christopher Kulla and Marcos Fajardo. Importance sampling techniques for path tracing in participating media. In *Computer graphics forum*, volume 31, pages 1519–1528. Wiley Online Library, 2012.
- [Lam60] Johann Heinrich Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Klett, 1760.
- [LMAK00] P Lecocq, S Michelin, D Arques, and A Kemeny. Mathematical approximation for real-time lighting rendering through participating media. In *Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on*, pages 400–401. IEEE, 2000.
- [Max63] J Clerk Maxwell. A dynamical theory of the electromagnetic field. *Proceedings of the Royal Society of London*, pages 531–536, 1863.

- [med] Clouds picture. <http://mev.fopf.mipt.ru>. Accessed: 2018-11-21.
- [Now18] Derek Nowrouzezahrai. Ecse 446/546: Realistic and advanced image synthesis. *McGill University*, September 2018.
- [Pix17] Pixar Animation Studios. *Coco*. [Blu-Ray], 2017.
- [PP09] Vincent Pegoraro and Steven G Parker. An analytical solution to single scattering in homogeneous participating media. In *Computer Graphics Forum*, volume 28, pages 329–335. Wiley Online Library, 2009.
- [rena] Disney’s new Production Renderer. <https://www.fxguide.com/featured/disneys-new-production-renderer-hyperion-yes-disney/>. Accessed: 2018-11-22.
- [renb] How Pixar made Monsters University, its latest technological marvel. <https://venturebeat.com/2013/04/2/the-making-of-pixars-latest-technological-marvel-monsters-university/>. Accessed: 2018-11-22.
- [renc] New technology revs up Pixar’s ‘Cars 2’. <https://www.cnet.com/news/new-technology-revs-up-pixars-cars-2/>. Accessed: 2018-11-22.
- [sta] Value of the global entertainment and media market from 2011 to 2021. <https://www.statista.com/statistics/237749/value-of-the-global-entertainment-and-media-market/>. Accessed: 2018-11-14.
- [Ste] Steve Lacey. Fog picture. Accessed: 2018-11-21.
- [SWZ96] Peter Shirley, Changyaw Wang, and Kurt Zimmerman. Monte carlo techniques for direct lighting calculations. *ACM Transactions on Graphics (TOG)*, 15(1):1–36, 1996.
- [UFK13] Carlos Ureña, Marcos Fajardo, and Alan King. An area-preserving parametrization for spherical rectangles. In *Computer Graphics Forum*, volume 32, pages 59–66. Wiley Online Library, 2013.
- [Vea97] Eric Veach. *Robust monte carlo methods for light transport simulation*. Number 1610. Stanford University PhD thesis, 1997.
- [WABG06] Bruce Walter, Adam Arbree, Kavita Bala, and Donald P Greenberg. Multidimensional lightcuts. *ACM Transactions on graphics (TOG)*, 25(3):1081–1088, 2006.

-
- [Wal16] Walt Disney Animation Studios. Moana. [Blu-Ray], 2016.
- [WBKP08] Bruce Walter, Kavita Bala, Milind Kulkarni, and Keshav Pingali. Fast agglomerative clustering for rendering. In *Interactive Ray Tracing, 2008. RT 2008. IEEE Symposium on*, pages 81–86. IEEE, 2008.
- [WFA⁺05] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P Greenberg. Lightcuts: a scalable approach to illumination. *ACM Transactions on graphics (TOG)*, 24(3):1098–1107, 2005.
- [wik] Explosion - Wikipedia. <https://upload.wikimedia.org/wikipedia/commons/c/c7/Explosions.jpg>. Accessed: 2018-11-21.