




Supplementary Document: Linearly Transformed Spherical Distributions for Interactive Single Scattering with Area Lights

Aakash KT¹ , Ishaan Shah¹ , P. J. Narayanan¹ 

¹International Institute Of Information Technology, Hyderabad (IIIT-H)

Abstract

This is the supplementary document showing the derivation of inverse cosine sampling and detailing our rendering algorithm.

CCS Concepts

• **Computing methodologies** → **Ray tracing; Reflectance modeling;**

1. Sampling Inverse Cosine

In this section we derive the sampling routine for sampling for a inverse cosine distribution i.e $1 - \cos \theta$. We use the CDF inversion technique [PJH16] to this.

Since the PDF is not dependant upon ϕ it can be sampled uniformly in range $[0, 2\pi]$. The PDF $p(\theta, \phi)$ is given by:

$$p(\theta, \phi) = \frac{1}{\pi} (1 - \cos \theta) \sin \theta.$$

The marginal PDF of $p(\theta)$ can is:

$$\begin{aligned} p(\theta) &= \int_0^{2\pi} p(\theta, \phi) d\phi \\ &= \int_0^{2\pi} \frac{1}{\pi} (1 - \cos \theta) \sin \theta d\phi \\ &= 2 \sin \theta (1 - \cos \theta). \end{aligned}$$

The CDF $P(\theta)$ is given as:

$$\begin{aligned} P(\theta) &= \int_0^\theta 2 \sin \theta (1 - \cos \theta) d\theta \\ &= 2 \left[\int_0^\theta \sin \theta d\theta - \int_0^\theta \sin \theta \cos \theta d\theta \right] \\ &= 2 \left[-\cos \theta \Big|_0^\theta + \frac{1}{2} \cos^2 \theta \Big|_0^\theta \right] \\ &= 2 \left[1 - \cos \theta + \frac{1}{2} \cos^2 \theta - \frac{1}{2} \right] \\ &= 1 - 2 \cos \theta + \cos^2 \theta. \end{aligned}$$

Given a random number ξ sampled uniformly from $[0, 1]$, the θ can be found as:

$$\theta = P^{-1}(\xi),$$

where $P^{-1}(\xi)$ is the inverse of the CDF $P(\theta)$ which can be found as follows:

$$\begin{aligned} \xi &= 1 - 2 \cos \theta + \cos^2 \theta \\ \xi &= (1 - \cos \theta)^2 \\ \pm \sqrt{\xi} &= (1 - \cos \theta) \\ 1 \pm \sqrt{\xi} &= \cos \theta \\ \cos^{-1}(1 \pm \sqrt{\xi}) &= \theta. \end{aligned}$$

Since, the argument of \cos^{-1} should be in the range $0 - 1$, $P^{-1}(\xi) = \cos^{-1}(1 - \sqrt{\xi})$. Hence, given two uniformly distributed random numbers ξ_1, ξ_2 in $[0, 1]$, we can sample a θ, ϕ as follows:

$$\begin{aligned} \phi &= 2\pi \xi_1 \\ \theta &= \cos^{-1}(1 - \sqrt{\xi_2}). \end{aligned}$$

2. Algorithm

ALGORITHM 1: Semi-analytic unshadowed single scattering

```

1 Def analyticInScatter(mei):
2   setLocalCoord(mei.wo) //  $\omega_o$  aligned with z-axis
3    $\bar{L}_s = 0$ 
4    $A_1, A_2 = \text{splitAtHorizon}(A)$  // Split area light A
5    $S_1, S_2 = \text{solidAngle}(A_1), \text{solidAngle}(A_2)$ 
6   /* Fetch LTSD matrices with amplitudes */
7    $M_s, a_u = \text{fetchLTSDUpper}(mei.g)$ 
8    $M_l, a_l = \text{fetchLTSDLower}(mei.g)$ 
9   /* Evaluate Eq. 21 */
10  if  $mei.g < 0$  then
11     $\bar{L}_s = a_u \cdot E(M_s^{-1}A_1) + a_l \cdot [S_2 - E(M_l^{-1}A_2)]$ 
12  else if  $mei.g \geq 0$  then
13     $\bar{L}_s = a_u \cdot [S_1 - E(M_s^{-1}A_1)] + a_l \cdot E(M_l^{-1}A_2)$ 
14   $\bar{L}_s = T(A) \cdot \bar{L}_s$  // Mul. by analytic
15  return  $\bar{L}_s$ 
16
17 Def airLight(ray, si, N):
18   $\bar{A} = 0$ 
19   $\Delta z = si.tmax / N$  // Equally divide ray extent
20   $Q_1, Q_2 = 0$ 
21  for  $s$  in  $0 \dots N$  do
22     $t = ray.o + s * ray.d$ 
23     $mei = \text{getMediumInteraction}(t, ray)$ 
24     $\bar{L}_s = \text{analyticInScatter}(mei)$ 
25     $Q_2 = T(t) \cdot \bar{L}_s$  // Transmittance from Eq. 5
26    if  $s \neq 0$  then
27       $\bar{A} += \frac{Q_1 + Q_2}{2} \Delta z$  // Eq. 15
28     $Q_1 = Q_2$ 
29  return  $\mu_s \cdot \bar{A}$ 
30
31 Def renderPixel(ray, N):
32   $si = \text{rayCast}(ray)$ 
33  /* Air-light with N quadrature samples
34  (Eq. 15) */
35   $\bar{L} = \text{airLight}(ray, si, N)$ 
36  if  $si.hit$  then
37    /* LTC & mult. by analytic
38    transmittance (Eq. 10) */
39     $\bar{L}_c = \text{analyticSurface}(si) \cdot T(A)$  // A is the area
40    light
41     $\bar{L} += T(si.t) \cdot \bar{L}_c$ 
42  return  $\bar{L}$ 

```

The pseudo-code of our unshadowed semi-analytic single scattering is given in Alg. 1. The entry point is the function `renderPixel` on line 27, which accepts the pixel's ray and the number of quadrature samples N as input. We get the surface intersection for this ray in line 28. Next, the air-light integral is estimated using quadrature (line 29) and surface color is computed and added to it (lines 30-32).

The function `airLight` estimates the air-light integral by first equally dividing the ray's extent into N parts (line 16), resulting in N point samples. For each sample, we get its medium interaction and compute \bar{L}_s (lines 18-21). This is then multiplied by the transmittance between the camera and that sample (line 22), before

performing the quadrature sum. The obtained quantity after evaluating all samples is multiplied by μ_s , according to Eq. 15.

Next, the function `analyticInScatter` analytically evaluates \bar{L}_s at a single point in the medium. We set the local-coordinate frame such that ω_o aligns with the z-axis (line 2). We then split the solid angle of the area light at the horizon (lines 4-5). The LTSD matrices are fetched using the value of g , along with their amplitudes (lines 6-7). We then analytically evaluate the in-scattered radiance using M_s and M_l according to Eq. 21. Finally, we multiply the computed value by the analytic transmittance towards the area light (Eq. 10) and return (lines 12-13).

References

- [PJH16] PHARR M., JAKOB W., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*, 3 ed. Cambridge, MA, 2016. 1