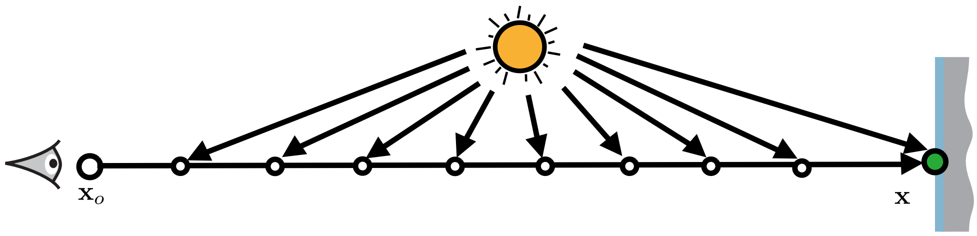
0.1 Equi-angular Volumetric Importance Sampling



Volumetric single scattering traces rays that start from the eye (*xo*), goes through the participating media, considers the incident radiance due to direct illumination but ignores incident radiance due to multiple scattering, finally hits a point at some surface in the scene (*x*).

Overview Volumetric single scattering is a method for rendering images where there are participating media in the scene, such as smoke, fog, fire, etc. There are three main processes that affect the distribution of radiance in an environment with participating media: absorption, emission, and scattering. Scattering is our topic of this section. The characteristics of all of these properties may be homogeneous or inhomogeneous. Homogeneous properties are constant throughout a given spatial extent, while inhomogeneous properties may vary throughout space. In this section, we focus on homogeneous isotropic media.

The volume rendering equation [CHA50] gives a formula for the radiance L seen at a position x when looking along a direction ω as the sum of two terms:

$$L(\mathbf{x}, \omega) = T(s)L_s(\mathbf{x}_s, \omega) + L_v(\mathbf{x}_s, \omega), \tag{1}$$

The surface radiance L_s defined by the familiar rendering equation [Kajiya86] is attenuated by transmittance T along the ray up to the distance s to the nearest surface intersection. The transmittance function is defined by the integral of the extinction coefficient s_t along the ray:

$$T(t) = exp(-\int_0^t \sigma_t(\mathbf{x}_w)dw), \tag{2}$$

To simplify the notation we write T as a univariate function, but it implicitly depends on the parameterized position $x_t = x+tw$ for any distance $t > 0$. Finally, light scattered inside the volume is accounted for by the volumetric radiance L_v:

$$L_v(\mathbf{x}, \omega) = \int_0^s T_r(\mathbf{x}_t)\sigma_t(\mathbf{x}_w)L_s(\mathbf{x}_t, \omega)dt, \tag{3}$$

where σ_t is the medium's scattering coefficient and r is the normalized phase function. The key difficulty in solving Equation 3 efficiently lies in the fact that the integration along the line contains two nested integrals: one for evaluating transmittance, the other for evaluating incoming light.

Basic Idea At each pixel, there are two methods for solving Equation 3, Riemann sum approximation and Monte Carlo integral estimation. The ray marching algorithm use Riemann sum to approximate the integral. By setting a step size, samples are distributed evenly on the primary ray between camera and the closest intersection point, both the transmission and lighting terms can be evaluated in lockstep. Every sample is used for solving another Monte Carlo to obtain the contribution of direct illumination. However, it is inefficient to form connections from so many samples to light sources.

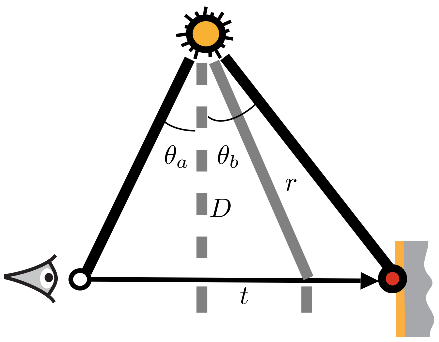
We solve Equation 3 by Monte Carlo estimation:

$$L_v(\mathbf{x}, \omega) = \frac{1}{N}\sum_{i=1}^{N} T_r(\mathbf{x}_{t,i})\sigma_t(\mathbf{x}_{t,i})L_s(\mathbf{x}_{t,i}, \omega_i)/pdf, \tag{4}$$

We consider the simpler problem of single scattering from point lights in a homogeneous medium, then L_s in Equation 4 would be

$$L_s(\mathbf{x}, \mathbf{z}) = \frac{\Phi}{4\pi} f_r(\mathbf{x}, \mathbf{p}, \mathbf{z})V(\mathbf{x}, \mathbf{p})\frac{cos\theta_i}{d^2}, \tag{5}$$

where $\mathbf{z}$ is the intersection point on surface, $\mathbf{p}$ is point light position, and Φ is radiant intensity, d is the distance between $\mathbf{z}$ and$\mathbf{p}$. Instead of sampling uniformly along primary ray, distance sampling generates samples proportional to $T_r(\mathbf{x}_{t,i})$ term, so that there will be more samples near the camera, where transmittance is greater. Equi-angluar sampling [Kulla12] generates samples proportional to the $\frac{1}{d^2}$ term, so that there will be more samples near the light source along the primary ray.



Single scattering from a point light source origin is at the orthogonal projection of the light onto the ray. Therefore the angle θ_a is negative.

Figure 2 describes the involved parameters. We reparameterize t so the origin is at the orthogonal projection of the light onto the ray. This change modifies the integration bounds a and b (which can be negative) and adds an extra term D which is the distance between the real ray origin and the new one.

We use normalized pdf that is proportional to the $\frac{1}{d^2}$ term, associated sampling function over the interval $[a, b]$ ($\mathbf{x}_i$ is a random number in $[0, 1)$).

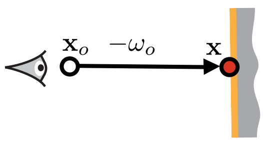
$$pdf(t) = \frac{D}{(\theta_b - \theta_a)(D^2 + t^2)}, \tag{6}$$

$$t(\mathbf{x}_i) = Dtan((1 - \mathbf{x}_i)\theta_a + \mathbf{x}_i\theta_b), \tag{7}$$

$$\theta_x = tan^{-1}(D/x), \tag{8}$$

Equation 7 reveals that this technique linearly interpolates the angle between the bounds of integration. We can generalize this result to spherical light sources by noticing that the solid angle subtended by the sphere also varies as $\frac{1}{d^2}$.

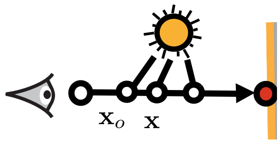
Algorithm Details For the purposes of our implementation, we will assume that a ray-tracer is used here, where rays are traced to compute (nearest-surface) object intersections. To begin, we assume that camera rays are generated and traced through each pixel. For each of these rays, we calculate the color $L_s(\mathbf{x}, \omega)$ at the nearest surface point $\mathbf{x}$ intersected by the ray, and the transmittance $T(\mathbf{x})$ from the camera to $\mathbf{x}$. Then we calculate the single scattering term using Monte Carlo estimation according to Equation 4. First generating samples using equi-angular sampling. For each sample on the camera ray, we calculate the



Step *i*: Rays are traced through each pixel, hit a nearest surface point.

color of the current sample, phase function, and transmittance between point light position and current sample, etc, divided by pdf. Finally, the average of all samples is used as the final value for L_v of the current pixel.

This procedure iterates through all pixels to get the final image, and anti-aliasing could also be used to improve the results.



Step *ii*: Generate sample proportional to $\frac{1}{d^2}$, calculate direct illumination from light for each sample.

Implementation Details The pseudocode below assumes that we have access to a ray-tracer that can `intersect` `rays` with the scene geometry and return `surface` intersection data. The `shade` function returns an RGB color corresponding to the 1-sample MC estimate, and so it should (ideally) be called and averaged *several times* per pixel. The function `getTransmittance` can return the transmittance between two given points using Equation 2. Phase Function is not explicitly written here, in implementation we use pf to be $\frac{1}{4\pi}$ for homogeneous isotropic media.

```
void sampleEquiAngular(
   float u,
   float maxDistance,
   vec3 rayOrigin,
   vec3 rayDir,
   vec3 lightPos)
{
   // get the closest point to light along the ray
   float delta = dot(lightPos - rayOrigin,
      rayDir);

   // get distance this point is from light
   float D = length(rayOrigin + delta*rayDir -
      lightPos);

   // get angle of theta_a and theta_b
   float thetaA = atan(0.0 - delta, D);
   float thetaB = atan(maxDistance - delta, D);

   // take sample
   float t = D*tan(mix(thetaA, thetaB, u));
   dist = delta + t;
   pdf = D/((thetaB - thetaA)*(D*D + t*t));

}
```

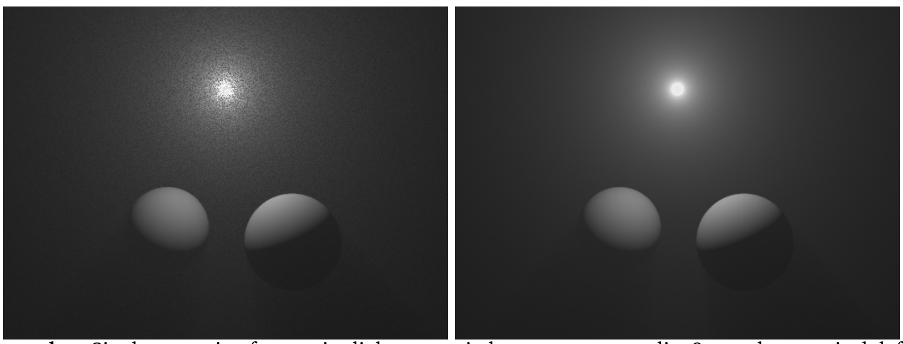
```
rgb volumetricSingleScatter(
   vec3 x,
   int N,
   ...
   )
{
   // get intersect point
   rgb L_s = shade(x);

   // get transmittance from this point to camera
   float Tr1 = getTransmittance(x,CameraPos);

   // Monte Carlo
   for(i from 1 to N)
   {
     vec3 x_t = sampleEquiAngular(...);
     //get phase function pf, transmittance Tr2 and Tr3
     float Tr2 = getTransmittance(x_t,CamPos);
     float Tr3 = getTransmittance(x_t,LightPos);

     L_v += Tr2 * shade(sample) * pf * Tr3;
   }
   L_v /=N;
   return L_s + L_v;
}
```

The code in the left shows the equi-angular sampling procedure, while the right side is volumet-



Rendering results : Single scattering from point light source in homogeneous media, 8 samples per pixel, left is distance sampling, right is equi-angular sampling, each rendered in thirty seconds.

ric single scattering.. Our rendering results all assume fully diffuse surfaces and homogeneous isotropic media, where phase function $pf = \frac{1}{4\pi}$, and $\sigma_a = 0.001$, $\sigma_s = 0.003$.

Our rendering results shows distance sampling and equi-angular sampling from a point light source. As the samples approach light source, $1/(d^2)$ starts to dominate, but transmittance is bounded by 1. Distance sampling focuses most samples close to the camera, missing the visually important light source, therefore there are more noise in results showed in the left.

Volumetric Scattering Reference

[Kajiya86] James T. Kajiya. *The Rendering Equation*. Proceedings of the 13th annual conference on Computer graphics and interactive techniques (SIGGRAPH). 1986

[Kulla12] Kulla, Christopher, and Marcos Fajardo.*Importance sampling techniques for path tracing in participating media*. Computer Graphics Forum. Vol. 31. No. 4. Blackwell Publishing Ltd, 2012.

[CHA50] Chandrasekhar S. *Radiative Transfer*. Clarendon[J]. 1950.