

# Free Path Sampling in High Resolution Inhomogeneous Participating Media

Dezeming Family

2023 年 10 月 29 日

DezemingFamily 系列文章和电子书**全部都有免费公开的电子版**，可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列电子书，可以从我们的网站 [<https://dezeming.top/>] 找到最新的版本。对文章的内容建议和出现的错误也欢迎在网站留言。

## 目录

<b>一 介绍</b>	<b>1</b>
<b>二 Light Attenuation and Free Path Sampling</b>	<b>1</b>
<b>三 The New Method</b>	<b>1</b>
3 1 Free path sampling with virtual particles . . . . .	1
3 2 Piece-wise polynomial upper-bound . . . . .	1
3 3 算法细节 . . . . .	2
<b>参考文献</b>	<b>2</b>

## 一 介绍

该方法基于将“虚拟”材质或粒子混合到介质中的概念，将消光系数增加到一个函数，以便可以以直接的方式对自由路径进行采样。

选择虚拟材质以使其修改体 density 但不改变 radiance。我们通过超体素的低分辨率网格来定义真实粒子和虚拟粒子的总消光系数，超体素比定义介质的真实体素大得多。

所提出的方法的计算复杂性仅取决于超体素网格的分辨率，并且不会随着超体素尺度以上的分辨率而增长。该方法对于渲染大的、低密度的、异质的体特别有效，否则，这些体应该由极高分辨率的体素网格定义，这将导致平均自由路径长度将穿过许多体素。

我们的目标是推广 Woodcock 跟踪，并使其高效（即使是对于最大消光值与大多数点的消光值相差甚远的参与介质）。本文提出了一种有效的体自由路径采样方法，其中我们不仅有体素的密度值，而且有一个自由路径采样简单的上界函数。如果体可用作高分辨率体素阵列，则可以在渲染开始前计算边界函数。在由程序生成的体的情况下，可以直接从程序定义中获得边界函数，而无需生成高分辨率和非常大的体素阵列。

## 二 Light Attenuation and Free Path Sampling

### Ray marching

均匀介质根据概率密度采样，生成的  $s$  是符合 optical depth 值  $\tau$  的：

$$P(s) = 1 - \exp(-\tau(0, s)). \quad (3)$$

$$r = P(s) \Leftrightarrow -\log(1 - r) = \tau(0, s). \quad (4)$$

非均匀介质的 RayMarching:

$$\sum_{i=0}^{n-1} \sigma_i(\vec{p}(i \Delta s)) \Delta s \leq -\log(1 - r) < \sum_{i=0}^n \sigma_i(\vec{p}(i \Delta s)) \Delta s. \quad (5)$$

### Woodcock tracing

## 三 The New Method

为了解决这个问题，我们通过添加虚拟的“材质”或粒子来修改体，使总密度遵循一个简单的函数。有人可能会认为，修改材质密度也会改变体内部的光辐射，从而导致渲染解决方案失真，这显然是不可取的。幸运的是，如果虚拟材料的另外两个自由性质，即反照率和相位函数得到了适当的定义，情况就不一定如此。如果虚拟粒子在散射过程中不改变光子的能量和方向，它们就不会改变介质内部的辐射。如果虚拟粒子的反照率为 1，并且其相位函数为 Dirac-delta，则满足这一要求，因为在这种情况下，与虚拟粒子的碰撞既不会改变光子的能量，也不会以概率 1 改变光子的方向，因此虚拟材料不会影响光的辐射。

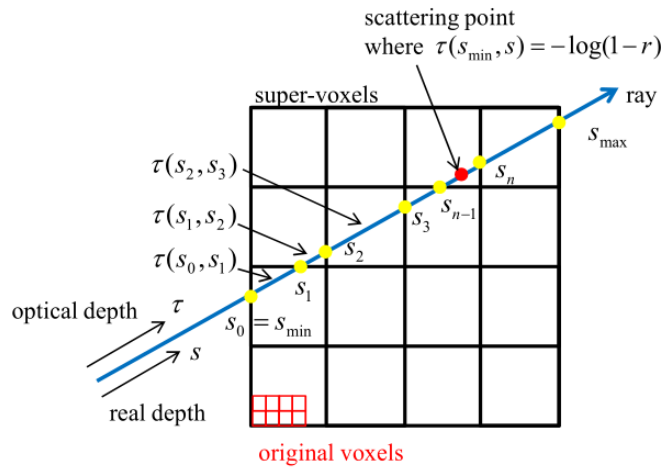
### 3 1 Free path sampling with virtual particles

这里还没有介绍超体素网格，因此就是一般的 Woodcock 追踪技术。

### 3 2 Piece-wise polynomial upper-bound

假设上限函数是定义在一个低分辨率网格上的分段多项式，然后去采样上限函数（毕竟其实在自由路径采样中其实本质也是采样上限函数，然后根据概率来判断是否会发生散射）。

由于上限函数被假设为是分段多项式，所以可以解析得求解来得到。虽然这里也是逐段计算，但不同于 Ray Marching，每段的长度与穿过这个超网格的点有关：



**Figure 4: 3D DDA algorithm to visit super-voxels.**<sup>1891</sup>

执行 3D DDA 算法，用来寻找散射点在哪个超体素网格中。

每个超体素中的  $\sigma_{max}$  都是分段多项式函数，与所在超体素的八个顶点的值相关。

### 3.3 算法细节

算法有两层大循环，外层判断是否发生了真实散射事件，内层是对一个超体素网格的 3D DDA 遍历过程。

结束内层循环以后，就找精确的散射位置，这里采用的是 SolvePolyInt 函数，大概是一种夹逼方法（从射入和射出点两头逼近找散射位置），然后判断该散射位置是否是零散射。

这个过程中，比较复杂的是内部循环，根据在超体素中的射入和射出点计算对最大消光值的积分，该过程需要用到不少初高中的多项式知识，比较麻烦，而且该过程相对来说比较耗时。而且你还得采样超体素的八个顶点再做插值，这个过程也挺耗费时间的。

### 参考文献

- [1] Hofmann N, Martschinke J, Engel K, et al. Neural denoising for path tracing of medical volumetric data[J]. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2020, 3(2): 1-18.