吸收发射方程和编程描述

Dezeming Family

2022 年 5 月 9 日

DezemingFamily 系列书和小册子因为是电子书，所以可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列书，可以从我们的网站 [https://dezeming.top/] 找到最新版。对书的内容建议和出现的错误欢迎在网站留言。

目录

一 基本介绍 **1**

二 离散化求解积分 **1**

2 1 分段近似 1

2 2 组合 2

三 积分校正 **3**

3 1 积分校正的原理与意义 3

3 2 关联颜色与非关联颜色 4

3 3 基于 jittering 的 Monte Carlo 方法 4

参考文献 **4**

一 基本介绍

本文是对 RayCasting 算法的一个总结和编程上的描述，对原理部分不会介绍太多，具体的体渲染方法可以参考《PBRT 系列 20-专业知识理论与代码实战-渲染概率与采样》以及《辐射传输方程的路径积分表示》。

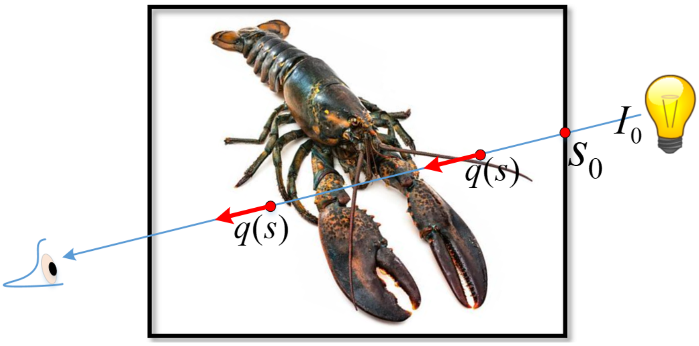
对于辐射传输方程，如果只考虑沿着一条采样路径的吸收项和发射项，就能得到：

$$\frac{dI(s)}{ds} = -\kappa(s)I(s) + q(s) \tag{一.1}$$

其中，$\kappa(s)$ 表示当前位置的衰减系数，$q(s)$ 表示当前位置发出的光亮度，$I(s)$ 表示积累到当前位置得到的光亮度。写成积分式为：

$$I(D) = I_0 e^{-\int_{s_0}^{D}\kappa(t)dt} + \int_{s_0}^{D} q(s)e^{-\int_{s}^{D}\kappa(t)dt}ds \tag{一.2}$$

其中，I_0 表示光沿着积分路径，在背面进入体空间 $s = s_0$ 的时候的亮度。可以看到，越是靠后的体素，发出的光在到达相机后衰减越多。



为了简化起见，设：

$$\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(t)dt \tag{一.3}$$

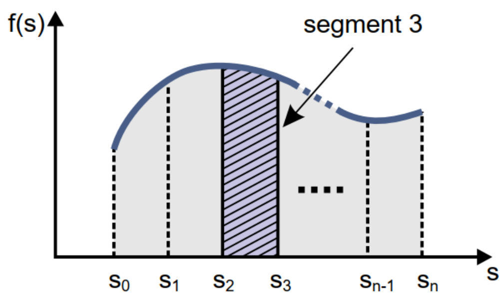
$$T(s_1, s_2) = e^{-\tau(s_1, s_2)} \tag{一.4}$$

$$\Longrightarrow I(D) = I_0 T(s_0, D) + \int_{s_0}^{D} q(s)T(s, D)ds \tag{一.5}$$

二 离散化求解积分

2 1 分段近似

尽管可以使用蒙特卡洛方法来求解积分式，但一般还是倾向于离散化近似——构建分段函数。



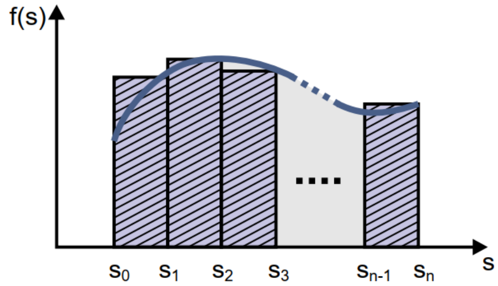
在第 $[s_{i-1}, s_i]$ 段里：

$$I(s_i) = I(s_{i-1})\underbrace{T(s_{i-1}, s_i)}_{T_i} + \underbrace{\int_{s_{i-1}}^{s_i} q(s)T(s, s_i)ds}_{c_i} \tag{二.1}$$

所以，$I(D)$ 可以写为：

$$\begin{aligned}I(D)=I(s_n)&=I(s_{n-1})T_n+c_n\\&=\big(I(s_{n-2})T_{n-1}+c_{n-1}\big)T_n+c_n\\&=\Big(\big(I(s_{n-3})T_{n-2}+c_{n-2}\big)T_{n-1}+c_{n-1}\Big)T_n+c_n\\&=\sum_{i=0}^{n}\Big(c_i\prod_{j=i+1}^{n}T_j\Big)\qquad c_0=I(s_0)\end{aligned}\tag{二.2}$$

采用黎曼和的形式，将一系列的矩形区域加起来：



每一段长度为 Δx：

$$\Delta x=\frac{D-s_0}{n}=s_i-s_{i-1},\quad 1\le i\le n\tag{二.3}$$

则第 i 段的穿透率和颜色值近似为：

$$\begin{aligned}T_i&\approx e^{-\kappa(s_i)\Delta x}\\c_i&\approx q(s_i)\Delta x\end{aligned}\tag{二.4}$$

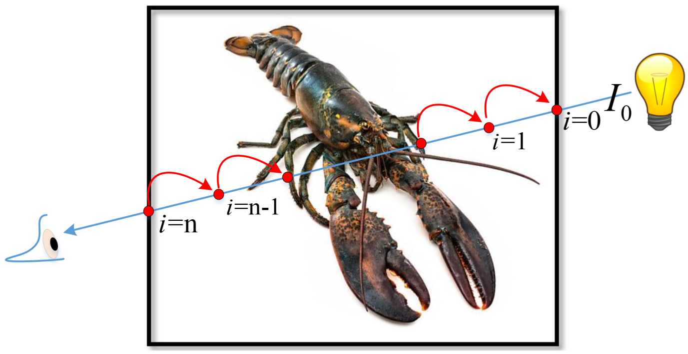
T_i 还比较容易理解，可是为什么 c_i 简化成了这个样子呢？这是因为我们默认在这一段中，$q(s)$ 没有被吸收，而是被下一段吸收了，所以只积累发射值。

2 2 组合

组合 (compositing) 其实就是求上述黎曼和，一共有两种模式，一种是从前到后，另一种是从后到前。

从前到后组合

从前到后就是采样光线从相机出发，逐步穿过体空间。定义 $i=n$ 为离相机最近的位置，$i=0$ 为体空间的背面：



设颜色值 C，有 RGB 三通道；体素的不透明度为 $\alpha_i=1-T_i$（就是通常我们认为的 RGBA 图像通道的 α 通道）。

迭代过程的初始化为：

$$\begin{aligned}\widehat{C}_n&=C_n\\\widehat{T}_n&=1-\alpha_n\end{aligned}$$

迭代过程（注意下标 i 比下标 $i+1$ 时间上在后面获取）：

$$\begin{aligned}\widehat{C}_i &= \widehat{C}_{i+1} + \widehat{T}_{i+1} C_i \\ \widehat{T}_i &= \widehat{T}_{i+1} T_i = \widehat{T}_{i+1}(1-\alpha_i)\end{aligned} \tag{二.5}$$

也就是说：

$$\begin{aligned}&\widehat{T}_n = 1-\alpha_n \Longrightarrow 1-\widehat{\alpha}_n = 1-\alpha_n \\ &\widehat{T}_{n-1} = 1-\widehat{\alpha}_{n-1} = \widehat{T}_n(1-\alpha_{n-1}) \\ &\widehat{\alpha}_i = \widehat{\alpha}_{i+1} + (1-\widehat{\alpha}_i)\alpha_i\end{aligned} \tag{二.6}$$

如果用 dst 下标来表示迭代中的“目标”值，用 src 表示采样中的每个位置的值，则可以写为下面的迭代过程：

$$\begin{aligned}C_{dst} &\longleftarrow C_{dst} + (1-\alpha_{dst})C_{src} \\ \alpha_{dst} &\longleftarrow \alpha_{dst} + (1-\alpha_{dst})\alpha_{src}\end{aligned}$$

从后到前组合

迭代过程的初始化为：

$$\begin{aligned}\widehat{C}_0 &= C_0 \\ \widehat{T}_0 &= 1-\alpha_0\end{aligned}$$

迭代过程（注意下标 i 比下标 $i-1$ 时间上在后面获取）：

$$\begin{aligned}\widehat{C}_i &= \widehat{C}_{i-1}(1-\alpha_i) + C_i \\ \widehat{T}_i &= \widehat{T}_{i-1}(1-\alpha_i)\end{aligned} \tag{二.7}$$

注意在该过程中，$\widehat{T}_i$ 并没有什么用处，因此可以忽略掉。

迭代过程写为：

$$C_{dst} \longleftarrow (1-\alpha_{src})C_{dst} + C_{src} \tag{二.8}$$

三　积分校正

3 1　积分校正的原理与意义

回忆一下穿透率 (transparency) 的计算：

$$\begin{aligned}\tau(s_i, s_i+\Delta x) &= \int_{s_i}^{s_i+\Delta x} \kappa(t)dt \\ T(s_i, s_i+\Delta x) &= e^{-\int_{s_1}^{s_1+\Delta x}\kappa(t)dt}\end{aligned} \tag{三.1}$$

当分段时，意味着该段的穿透率变为了：

$$T(s_i, s_i+\Delta x) = e^{-\int_{s_1}^{s_1+\Delta x}\kappa dt} = e^{-\kappa\Delta x}$$

对于不同的 Δx，例如 $\Delta_{d1}x$ 和 $\Delta_{d2}x$，穿透率分别为：

$$\begin{aligned}T_1 &= e^{-\kappa\Delta_{d1}x} \\ T_2 &= e^{-\kappa\Delta_{d2}x} \\ T_1 &= T_2^{\frac{\Delta_{d1}}{\Delta_{d2}}}\end{aligned} \tag{三.2}$$

对于不透明度来说，不同长度的两个间隔的不透明度关系是：

$$\alpha_1 = 1 - (1 - \alpha_2)^{\frac{\Delta_{d1}}{\Delta_{d2}}} \tag{三.3}$$

为什么需要透明度校正的原因，是体渲染中，由传输函数分配的体素属性一般都是不透明度值，而不是 κ。我们需要一开始拟定一个步长，α 值对应于该步长下的不透明度。对于不同长度的采样步长而言，我们需要这么一个关系式，来计算出不同步长下的不透明度。

颜色校正的关系：

$$\begin{aligned} c &= q\Delta x \\ c_1 &= c_2(\frac{\Delta_{d1}x}{\Delta_{d2}x}) \end{aligned} \tag{三.4}$$

3 2 关联颜色与非关联颜色

关联颜色 (associated colors) 表示已经提前乘以了不透明度，非关联颜色则没有预乘不透明度。

前面我们其实已经默认是预乘了不透明度的结果（不透明度小于 1，因此相当于变淡的颜色值）。我们现在推广到没有预乘不透明度的迭代方法。

从前到后组合为：

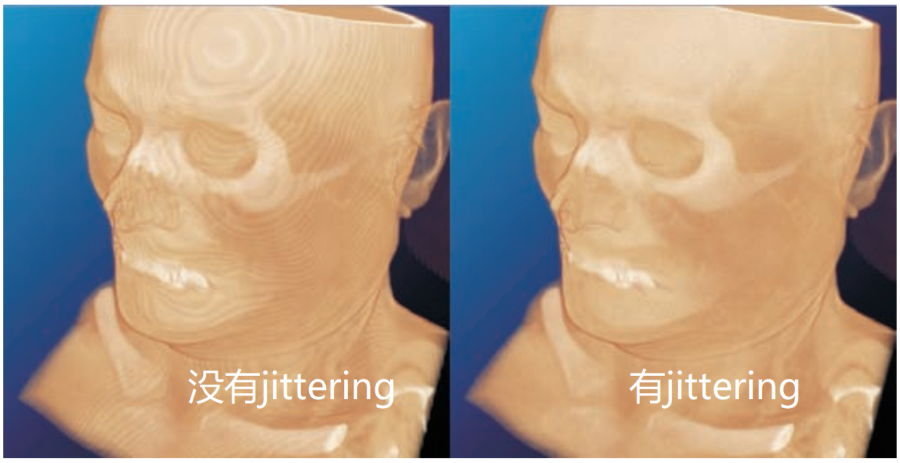
$$\begin{aligned} C_{dst} &\longleftarrow C_{dst} + (1 - \alpha_{dst})\big(\alpha_{src}C_{src}\big) \\ \alpha_{dst} &\longleftarrow \alpha_{dst} + (1 - \alpha_{dst})\alpha_{src} \end{aligned}$$

从后到前组合为：

$$C_{dst} \longleftarrow (1 - \alpha_{src})C_{dst} + \alpha_{src}C_{src} \tag{三.5}$$

3 3 基于 jittering 的 Monte Carlo 方法

传输函数本身也会给体数据增加高频性，使得渲染结果很容易出现“木纹”artifact。



jittering 就是在采样时，出发点往前偏移一个随机长度（不同像素偏移的长度是不同的），这样可以利用噪声来平衡 artifacts。

参考文献

[1] Engel K, Hadwiger M, Kniss J M, et al. Real-time volume graphics[M]//ACM Siggraph 2004 Course Notes. 2004: 29-es.