



南京大學

本科畢業論文

院 系 計算機科學與技術系

專 業 計算機科學與技術

題 目 高逼真半透明材質的渲染方法

年 級 2013 級 學 號 121140013

學生姓名 高 端

指導教師 1 徐 昆 職 稱 副教授

指導教師 2 過 洁 職 稱 講師

提交日期

摘要:

本文提出一种通过神经网络来拟合已有数据进行渲染的半透明材质渲染算法。我们首先回顾了真实感渲染的基本理论和体渲染的概念，并分类介绍了半透明材质渲染领域的主流算法，其中详细介绍了基于扩散理论的方法和基于预计算的方法。我们提出的方法基于前人提出的一种预计算方法，但该方法存在数据量过大以及拟合不太准确的情况，因此我们提出使用神经网络来实现更好的拟合效果。之后介绍了我们借助神经网络来拟合数据的思路策略以及具体的训练方法；我们实现了基于光线跟踪采样技术和层次化加速技术两种常用的半透明材质渲染框架，并在其上实现了我们的方法及其它几种经典的半透明材质渲染算法。最后，通过对拟合结果和渲染结果进行分析，证明我们的模型可以模拟更加准确的半透明效果。

关键词：半透明材质，次表面散射，BSSRDF，神经网络

Abstract

This paper introduces a translucent rendering algorithm than use neural network to fit simulated data. Firstly, we introduce the basic theory of photorealistic and volume rendering, then we introduce several major algorithms in translucent material rendering including diffusion theory based method and precomputed based method. We propose to fit data with neural network because previous precomputed work needs huge data and inaccurate. Secondly, we introduce how to use neural network to fit it well. Thirdly, we implement our method and several classic algorithm by both ray tracing and hierarchy acceleration framework. Lastly, we prove our model is more accurate by analysis fitting results and rendering results.

Keywords: Translucent materials, subsurface scattering, BSSRDF, neural network.

目录

- 1 绪论5
 - 1.1 简介5
 - 1.2 前人工作及研究现状5
 - 1.2.1 真实感渲染5
 - 1.2.2 半透明材质渲染15
 - 1.2.3 神经网络基本内容28
- 2 正文30
 - 2.1 研究目的和动机30
 - 2.2 研究内容31
 - 2.2.1 拟合策略31
 - 2.2.2 神经网络训练33
 - 2.2.3 半透明材质渲染的实现35
 - 2.3 实验方法和流程38
 - 2.3.1 拟合数据38
 - 2.3.2 渲染实现42
- 3 结论和讨论44
 - 3.1 拟合结果44
 - 3.1.1 神经网络训练误差44
 - 3.1.2 BSSRDF 随距离 r 变化曲线46
 - 3.1.3 二维投影图拟合结果49
 - 3.2 渲染结果53
 - 3.2.1 简单场景渲染结果53
 - 3.2.2 复杂几何体渲染结果56
 - 3.2.3 全局光照渲染结果57
 - 3.3 讨论和展望58
 - 3.3.1 不足58
 - 3.3.2 展望59
- 4 参考文献60
- 5 致谢62
- 6 附录63

1 绪论

1.1 简介

半透明材质是材质渲染中很重要的部分，比如常见的诸如皮肤、牛奶、果汁、大理石等材质都具有明显的半透明材质特性。



图1-1 半透明材质渲染图，材质为皮肤，该图片来自 [1] 中插图

半透明材质和其它材质的区别在于次表面散射是不可忽略的，次表面散射是指入射光线在介质表面除了发生表面散射外，还会进入介质内部发生多次的散射行为后从表面其它点出射。这种情况下，每个点的出射情况不仅和该点的入射光线有关，还与其周围的照明情况、几何形体有紧密的联系。因此，半透明材质的渲染是一项基础且重要的工作。

1.2 前人工作及研究现状

本章将从最基本的真实感渲染的概念开始，着重介绍半透明材质渲染领域的前人工作，并在此基础上对比分析这些工作的优劣势，最后简单介绍神经网络的基本原理等。

1.2.1 真实感渲染

渲染是从对 3D 场景的描述出发生成图片的过程，其中真实感渲染指渲染得到的图片和相机拍摄的照片几乎不可区分。真实感渲染除能给人产生沉浸式的逼真体验外，还提供了良定义的渲染系统质量的度量标准 [1] 。

本节将从基本的物理量定义出发，介绍用于描述物体表面散射的 BRDF 和 BSSRDF 及对应的渲染方程，之后介绍体渲染方法的基本概念，最后介绍其它有用的概念和技术。

1.2.1.1 基本概念

1. 一些基本概念

这里介绍图形学中使用到的辐射度学中的基本概念及其物理含义、定义式、单位以及相互关系。

- 辐射能量 (Radiant energy)

描述一定空间中所有的光子能量，用 Q 表示，单位是焦耳J。

光源会向场景中发射光子，每个光子有特定的波长并携带一定量的能量。

波长为 λ 的光子能量为： $Q = \frac{hc}{\lambda}$ ，其中 c 是光速， h 是普朗克常量

- 辐射通量 (Radiant flux)

描述单位时间内通过某面积区域的辐射能量，通常用 Φ 表示，单位是瓦特W， 也叫做辐射功率 (Radiant power)，其定义式：

$$\Phi = \lim_{\Delta t \rightarrow 0} \left(\frac{\Delta Q}{\Delta t} \right) = \frac{dQ}{dt}$$

其标量形式叫做辐射

- 辐射照度 (Irradiance) ¹

描述到达物体表面的单位面积内的辐射通量，通常用 E 表示，单位是 W/m^2 ，其定义式： $E = \frac{d\Phi}{dA}$

¹ 辐射照度和辐射出射度 (Radiant exitance) 定义式相同，区别在于辐射照度描述到达表面的单位面积光通量，而辐射出射度描述离开表面的单位面积辐射通量。

- 辐射出射度（Radiant exitance）

描述离开物体表面的单位面积内的辐射通量，通常用 M 表示，单位是 W/m^2 ，定义式： $M = \frac{d\Phi}{dA}$

- 辐射强度

描述单位立体角内通过的辐射通量，通常用 I 表示，单位是 W/sr , sr 是立体弧度单位，定义式： $I = \frac{d\Phi}{d\omega}$, 其中 ω 是方向向量

- 辐射亮度

辐射照度（及辐射出射度）描述了单位面积内的辐射通量，不涉及方向信息，而辐射亮度在此基础上加入方向信息，定义式

$L(p, \omega) = \lim_{\Delta\omega \rightarrow 0} \frac{\Delta E_{\omega}(p)}{\Delta\omega} = \frac{dE_{\omega}(p)}{d\omega}$, 其中, E_{ω} 代表垂直于方向 ω 的辐射照度。

也可以定义为： $L = \frac{d\Phi}{d\omega dA^{\perp}} = \frac{d^2\Phi}{dA \cos\theta d\omega}$, 其中, dA^{\perp} 代表投影面积, $\cos\theta$ 代表 ω 和表面法向量夹角的余弦。

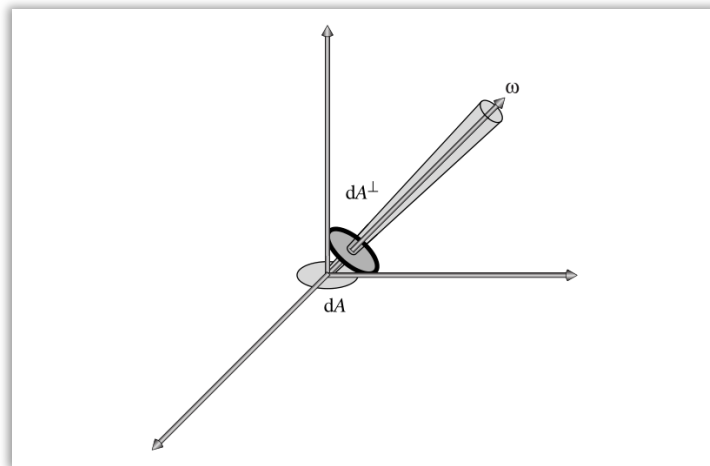


图1-2 辐射亮度 L 定义示意图，该图片来自 [1] 中插图

2. 表面散射

光线入射到物体表面后，会发生散射现象，通常对于不透明材质使用 BRDF（bidirectional reflectance distribution function，双向反射分布函数）

来描述，对于半透明材质则使用 BSSRDF（bidirectional scattering surface reflectance distribution function，双向表面散射分布函数）来描述 [2]。

BRDF 假定出射点和入射点为表面同一个点，因此忽略了次表面散射行为，对于不透明的材质而言，该近似误差很小且渲染效率很高，而对于次表面散射很多的半透明材质而言，必须使用一般化的 BSSRDF（出射点和入射点可以是不同的点）才可以准确建模。

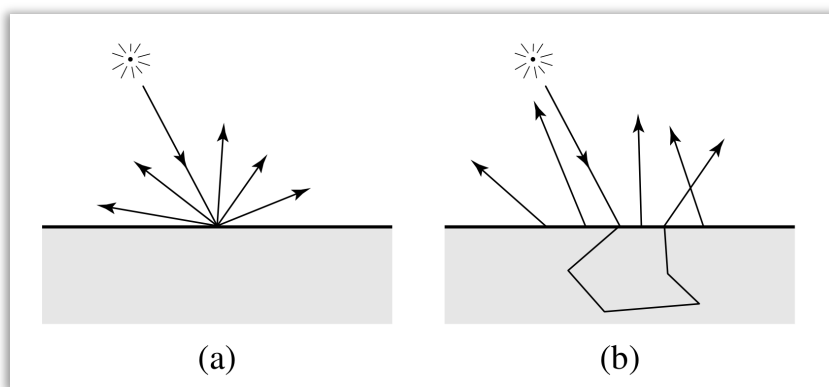


图1-3 BRDF（图 a）和 BSSRDF（图 b）对比

- BRDF

BRDF 描述了表面一点 p 处出射辐射亮度和入射辐射照度的比值,用符号 f_r 表示:

$$f_r(p, \omega_o, \omega_i) = \frac{dL_o(p, \omega_o)}{dE(p, \omega_i)} = \frac{dL_o(p, \omega_o)}{L_i(p, \omega_i) \cos \theta_i d\omega_i} \quad (1)$$

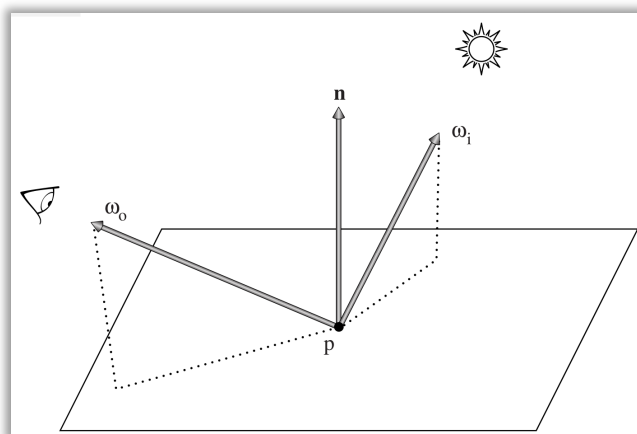


图1-4 BRDF 定义示意图，该图片来自 [1] 中插图

BRDF 满足两条重要性质：

1. 对称性: 交换入射、出射方向, BRDF 结果不变。

$$f_r(p, \omega_i, \omega_o) = f_r(p, \omega_o, \omega_i)$$

2. 能量守恒: 即反射的总能量小于等于入射的总能量。

$$\int_{\Omega} f_r(p, \omega_o, \omega') \cos\theta' d\omega' \leq 1$$

此外, 对于透射行为有类似的 BTDF (bidirectional transmittance distribution function, 双向透射分布函数) 进行描述, 二者合称 BSDF (bidirectional scattering distribution function, 双向散射分布函数), 用 $f(p, \omega_o, \omega_i)$ 表示。常见的 BRDF 模型包括 Lambert 漫反射模型、Phong 模型、微面元模型等, 这些模型均拥有形式简单的解析表达, 因此对其进行采样和渲染比较容易。

散射方程:

在 BRDF 定义的基础上, 可以计算某个方向的出射亮度:

$$L_o(p, \omega_o) = \int_{\Omega} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos\theta_i| d\omega_i \quad (2)$$

若上式的积分区域为整个球面, 则该方程叫做散射方程; 若积分区域为半球面, 则通常叫做反射方程。

- BSSRDF

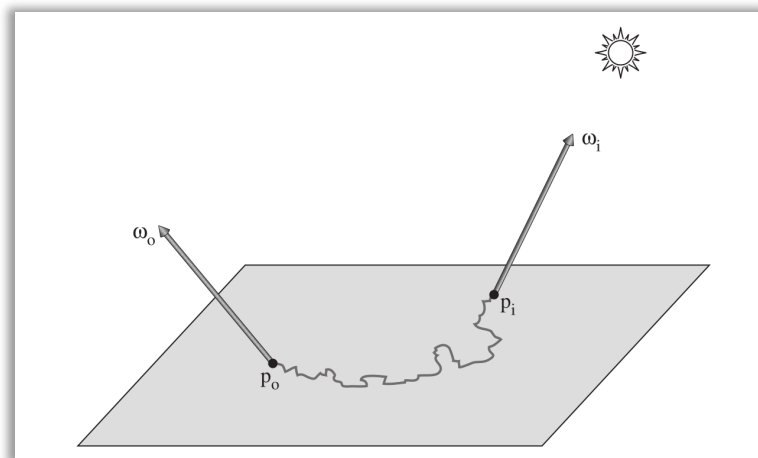


图1-5 BSSRD 定义示意图, 该图片来自 [1] 中插图

BSSRDF 描述了某一点 p_o 处出射辐射亮度和另一点 p_i 处入射辐射通量的比值,用 S 表示:

$$S(p_o, \omega_o, p_i, \omega_i) = \frac{dL_o(p_o, \omega_o)}{d\Phi(p_i, \omega_i)} \quad (3)$$

从公式的定义可以发现, S 和入射点位置和入射方向都有关系, 因此使用 BSSRDF 进行渲染, 需要同时对入射方向以及表面区域进行积分。另外, 由于其定义的复杂性, BSSRDF 并没有准确的解析式表示, 对其进行采样和渲染都较为困难。

一般化的散射方程:

$$L_o(p_o, \omega_o) = \int_A \int_{\Omega} S(p_o, \omega_o, p_i, \omega_i) L_i(p_i, \omega_i) |\cos \theta_i| d\omega_i dA \quad (4)$$

1.2.1.2 体渲染

光线在介质内的散射、吸收、反射等行为都可以通过光传输方程 (Radiant transfer equation, 简称 RTE) 来描述 [3]。而次表面散射可以看作是更微观尺度上的体散射, 因此也可以通过求解 RTE 来得到准确的结果。

下面介绍一些体渲染中的基本概念:

- 光线在介质中的三种行为
 - 吸收: 光能转换为其它形式的能量导致的辐射亮度的衰减;
 - 发射: 发光物体发射光子导致的辐射亮度的增加;
 - 散射: 光子和其它物体碰撞而导致的传输方向的改变。
 - ◆ 外散射: 散射行为使得光线偏离到不同的方向, 导致某个方向出射辐射亮度的减少。
 - ◆ 内散射: 散射行为使得光线集中到某个方向, 导致该方向的出射辐射亮度增加。

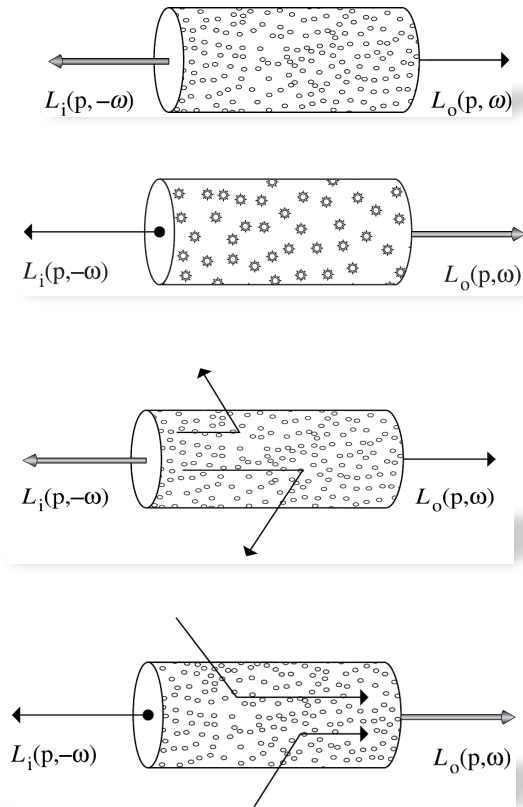


图1-6 光线在介质中几种行为示意图，该图片来自 [1] 中插图
从上到下依次为：吸收、发射、外散射、内散射

- 几个光学参数
 - 吸收系数 σ_a ：光线在介质中单位距离上被吸收的概率密度，其单位是 m^{-1} ，取值为任意正数。
 - 散射系数 σ_s ：光线在介质中单位距离上发生散射的概率密度，单位和取值范围同上。
 - 衰减系数 σ_t ：描述由于吸收以及外散射导致的出射辐射亮度的减少，定义为 $\sigma_t = \sigma_s + \sigma_a$ ，单位和取值范围同上。
 - 吸光度 α ：描述了散射占衰减的比例，定义为 $\alpha = \frac{\sigma_s}{\sigma_t}$ ，该物理量无量纲，取值范围是 $[0,1]$ 。
 - 平均自由程 mfp ：描述了光线在介质中和介质颗粒发生相互作用前的平均距离，定义为： $\text{mfp} = \frac{1}{\sigma_t}$ ，单位是 m 。
- 原项

原项描述了由于发射和内散射行为导致的单位距离内辐射亮度的增加，通常用 L_s 表示：

$$L_s(p, w) = L_e(p, w) + \sigma_s(p, w) \int_{\Omega} p(p, w_i, w_o) L_i(p, w_i) dw_i \quad (5)$$

上式中 σ_s 为散射系数，而 p 代表相函数，相函数描述了某一点处辐射量的角度分布，相函数满足归一化条件：即：

$$\int_{\Omega} p(p, \omega, \omega') d\omega' = 1$$

定义各向异性因子 g 来描述相函数的各向异性，

$$g = \int_{\Omega^2} p(-\omega, \omega') (\omega \cdot \omega') d\omega' = 2\pi \int_0^\pi p(-\cos\theta) \cos\theta \sin\theta d\theta$$

其取值为 0 代表各向同性，取值为正代表前向散射为主，为负

代表后向散射为主。

- RTE

结合原项和衰减因子，可以给出完整的光传输方程：

$$\frac{\partial}{\partial t} L_o(p + tw, w) = -\sigma_t(p, w) L_i(p, -w) + L_s(p, w) =$$

$$-\sigma_t(p, w) L_i(p, -w) + L_e(p, w) + \sigma_s(p, w) \int_{\Omega} p(p, w_i, w_o) L_i(p, w_i) dw_i \quad (6)$$

总结：体渲染方法通过对光线和介质颗粒的相互作用建模来准确计算光传输过程，但直接求解该方程用于半透明材质的渲染虽然很准确，但时间代价昂贵并且收敛极慢，一般仅作为参考标准使用。实用的半透明材质渲染方法本质上都是基于一定的假设来近似 RTE 的解，这些方法的具体介绍参见第二章。

1.2.1.3 其它内容

这里介绍一些实验中使用到的其它的技术或者概念。

- 蒙特卡罗积分

蒙特卡罗积分是求解积分的常用数值方法，一大优点是其收敛速度和被积函数的维度无关，因此特别适合于图形学中高维积分的计算。

对于积分 $\int_a^b f(x)dx$, 根据概率密度分布函数 $p(x)$ 获得 N 个采样 X_1, \dots, X_N , 则对应的蒙特卡罗估计量为:

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

容易证明, $E[F_N] = \int_a^b f(x)dx$, 因此可以采用该估计量代替原积分。

- 俄罗斯轮盘技术 (Russian roulette)

俄罗斯轮盘技术是一种根据每个采样的重要程度来增加蒙特卡罗估计量似然值的方法, 从而可以提高采样的效率。

基本思想是, 对每个采样, 以 p 的概率进行计算求解, 而 $1-p$ 的概率直接忽略, 得到的新的估计量:

$$F' = \begin{cases} \frac{F}{q}, & \xi < q \\ 0, & \xi \geq q \end{cases}$$

新的估计量的期望 $E[F'] = q \cdot \frac{E[F]}{q} + 0 \cdot (1 - q) = E[F]$.

实验中使用该技术来加快收敛。

- 物体表面的均匀采样

BSSRDF 采样算法的一大难点就在于对给定的出射点, 如何采样得到同一个物体上附近的入射点位置, 实验算法中采用了预先生成物体表面均匀采样的方式代替在每个出射点处重新采样。

平面内的均匀采样较为简单, 可以使用泊松圆盘采样来生成, 而 Bowers 等人 2011 年的工作提出了对任意几何形体进行均匀采样的算法 [4]。

泊松圆盘采样指生成的采样点随机分布的同时满足最小距离大于某个给定值。即在曲面 \mathcal{M} 上生成采样点集 \mathcal{S} , 满足 $d(s_i, s_j) \geq r, \forall s_i, s_j \in \mathcal{S}$, r 为给定的最小距离, d 是距离度量。

Bowers 等人的算法中首先生成均一的、足够大的点集作为初始结果，然后将这些点划分到不同的网格单元中，最终在每个网格单元内进行泊松圆盘采样来得到最终的结果。



图1-7 复杂几何体的泊松圆盘采样示意图，该图片引用自[4]论文插图。

- 菲涅尔反射

菲涅尔方程描述了光在不同介质传播时发生的反射和折射现象。形象的来讲，当我们垂直看向水面的时候反射光较弱，因此水看起来很透明，而当我们看很远处的水面时（视线方向和水面法向量夹角很大）反射很强。

对于电解质材质（dielectric material），其菲涅尔反射率 F_r 和透射率 F_t 定义如下：

$$r_{\parallel} = \frac{\eta_t \cos \theta_i - \eta_i \cos \theta_t}{\eta_t \cos \theta_i + \eta_i \cos \theta_t}$$

$$r_{\perp} = \frac{\eta_i \cos \theta_i - \eta_t \cos \theta_t}{\eta_i \cos \theta_i + \eta_t \cos \theta_t}$$

$$F_r = \frac{1}{2}(r_{\parallel}^2 + r_{\perp}^2)$$

$$F_t = 1 - F_r$$

菲涅尔漫反射反射率（Fresnel diffuse reflectance, 记作 F_{dr} ）,其定义为：

$$F_{dr} = \int_{2\pi} F_r(\eta, \vec{n} \cdot \vec{\omega}') (\vec{n} \cdot \vec{\omega}') d\omega'$$

一般情况下都使用其有理近似[6]:

$$F_{dr} = -\frac{1.440}{\eta^2} + \frac{0.710}{\eta} + 0.668 + 0.0636\eta$$

其它材质的菲涅尔反射率的计算大多需要使用近似公式进行，不再赘述。

1.2.2 半透明材质渲染

半透明材质的渲染方法可以大致分为：基于蒙特卡罗光线跟踪直接求解 RTE、基于扩散（Diffusion）近似的方法、基于预计算的方法。

本节将分别介绍这些内容，其中基于扩散的方法是半透明材质渲染的主流方法，因此将会重点介绍几种该类别的算法。

(1) 光线跟踪方法

首先介绍基于光线跟踪的方法，该方法利用蒙特卡罗光线跟踪技术求解 RTE（参见上一节体渲染的介绍），可以准确求解半透明材质，缺点在于收敛慢、时间代价昂贵。

RTE 的形式如下：

$$\begin{aligned}\frac{\partial}{\partial t} L_o(p + t\omega, \omega) &= -\sigma_t(p, \omega)L_i(p, -\omega) + L_s(p, \omega) \\ &= -\sigma_t(p, \omega)L_i(p, -\omega) + L_e(p, \omega) \\ &\quad + \sigma_s(p, \omega) \int_{\Omega} p(p, \omega_i, \omega_o)L_i(p, \omega_i)d\omega_i\end{aligned}$$

通过蒙特卡罗光线跟踪算法求解 RTE 是半透明材质渲染的最基本的方法。根据介质的匀质与否（介质内部所有区域拥有相同的光学参数则该介质是匀质的（homogeneous），否则是异质的（heterogeneous）），存在不同的相函数模型以及对应的采样例程。

1.2.2.1 基于扩散理论的方法

扩散理论（Diffusion theory）给我们提供了将复杂的 RTE 转换为更简单的扩散方程（Diffusion equation）的方法，而在介质匀质、高吸光度的情况下，扩散方程的解可以作为 RTE 的解。而常见的渲染模型中，均利用扩散理论的解以及边界条件来给出 BSSRDF 的近似表达式，再代入一般化的散射方程（公式（4））来计算。

下面具体介绍每种模型中应用的扩散理论以及每种模型用来简化 BSSRDF 表示的假设，并介绍每种模型的渲染方法。¹

- **2001 年 Jensen 等提出的对偶极子（Dipole）模型 [7]**

该模型受医学物理学领域中使用扩散模型预测高散射材质光学性质工作 [5] 的启发，首次将扩散理论引入计算机图形学半透明材质的渲染中，并给出单次散射（single scattering）求解方法、增加对任意几何体的支持以及提出完整的渲染算法。

该模型首先将 BSSRDF 分解为单次散射 S_1 和多次散射 S_d 部分：

$$S(p_i, \omega_i; p_o, \omega_o) = S_1(p_i, \omega_i; p_o, \omega_o) + S_d(p_i, \omega_i; p_o, \omega_o)$$

单次散射

其中，单次散射可以利用光线跟踪等方法进行求解：

$$\begin{aligned} L_o^{(1)}(p_o, \omega_o) &= \int_A \int_{2\pi} S_1(p_i, \omega_i; p_o, \omega_o) L_i(p_i, \omega_i) (\vec{n} \cdot \vec{\omega}_i) d\omega_i dA \\ &= \sigma_s \int_{2\pi} F_t(\eta, \vec{\omega}_o) F_t(\eta, \vec{\omega}_i) p(\omega'_i, \omega'_o) \int_0^\infty e^{-\sigma_{tc}s} L_i(p_i, \omega_i) ds d\omega_i \end{aligned}$$

$$\text{其中, } \sigma_{tc} = \sigma_t(p_o) + G\sigma_t(p_i), G = \frac{|n_i \cdot \omega'_o|}{|n_i \cdot \omega'_i|}$$

多次散射

¹ 标准对偶极子模型是基于扩散理论渲染半透明材质的基础，会完整介绍其思路和推导；快速对偶极子模型、多偶极子模型、量子对偶极子模型等介绍其主要思想和结果；光子扩散方法、方向对偶极子模型简略介绍其改进。

该方法基于一个关键性的观察：在高散射（指吸光度大）的介质中，出射光线的分布趋向于各向同性，即使是入射光分布、相函数等是高度各向异性的情况下该观察仍然成立。该现象成立的原因在于散射行为会模糊光线的分布，因此随着散射次数的增加，光线的分布趋向于各向同性。

下面介绍扩散方程的推导以及如何引入 Dipole 对其进行求解的具体过程。

扩散方程推导

在扩散理论的假设下，辐射亮度可以使用其球面矩来分解为两项之和，即使用零阶球面矩和一阶球面矩之和近似辐射亮度。

其零阶球面矩恰好是积分通量率（fluence rate）： $\Phi(p) = \int_{S^2} L(p, \omega) d\omega$

其一阶球面矩是向量辐射照度： $\vec{E}(p) = \int_{S^2} \omega L(p, \omega) d\omega$

因此近似后的辐射亮度表达式为：

$$L(p, \omega) = \frac{1}{4\pi} \Phi(p) + \frac{3}{4\pi} \vec{\omega} \cdot \vec{E}(p) \quad (7)$$

将该式代入 RTE，并要求等式两边的零阶矩和一阶矩相等，得到：

$$\begin{cases} \text{div } \vec{E}(p) = (-\sigma_t' + \sigma_s') \Phi(p) = -\sigma_a \Phi(p) + Q_0(p) \\ \frac{1}{3} \nabla \Phi(p) = -\sigma_t' \vec{E}(p) + Q_1(p) \end{cases}$$

其中, $Q_i(p)$ 代表介质发射光 $L_e(p, \omega)$ 的 i 阶矩。

上述两式联立可以消去变量 E ，得到关于零阶矩 $\Phi(p)$ 的方程，该方程就是经典扩散方程：

$$\frac{1}{3\sigma_t'} \nabla^2 \Phi(p) = \sigma_a \Phi(p) - Q_0(p) + \frac{1}{\sigma_t'} \nabla \cdot Q_1(p) \quad (8)$$

一个常见的简化是假定介质中各方向发射光是均匀的，也就是 $Q_1(p) = 0$ ：

$$D \nabla^2 \Phi(p) = \sigma_a \Phi(p) - Q_0(p) \quad (\text{扩散系数 } D = \frac{1}{3\sigma_t'}) \quad (9)$$

基于 Dipole 求解扩散方程

假定表面是半无限平面，并假设光源是点光源的情况下，扩散方程的解为：

$$\phi(p) = \frac{\Phi}{4\pi D} \left(\frac{e^{-\sigma_{tr}r(p)}}{r(p)} \right)$$

Φ 是点光源的辐射功率, r 是到点光源的距离, $\sigma_{tr} = \sqrt{3\sigma_a\sigma_t}$ 是有效传输系数。

该方法利用在平面上下各放置一个点光源, 构成对偶极子 (dipole) 的方式来满足边界条件: 其中真实光源放置在平面下距离为 $z_r = \frac{1}{\sigma'_t}$ 的位置, 虚拟光源放置在平面上方距离为 $z_v = z_r + 4AD$ 的位置 (其中, $A = \frac{1+F_{dr}}{1-F_{dr}}$, F_{dr} 的定义参见前一节)。

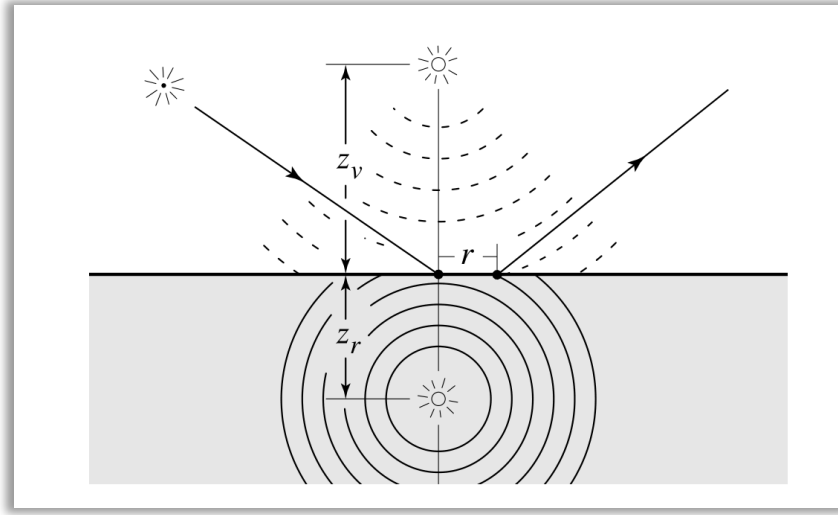


图1-8 Dipole 示意图, 该图片来自 [7] 插图

在该设置下, 可求解:

$$\phi_p = \frac{\Phi}{4\pi D} \left(\frac{e^{-\sigma_{tr}d_r}}{d_r} - \frac{e^{-\sigma_{tr}d_v}}{d_v} \right) \quad (10)$$

其中, d_r, d_v 代表 p 点距离真实光源、虚拟光源的距离。

进一步, 可得到漫反射反射率 R_d :

$$R_d = -D \frac{\vec{n} \cdot \vec{\nabla} \phi(p)}{d\Phi_i} = \frac{\alpha'}{4\pi} \left[z_r(\sigma_{tr}d_r + 1) \frac{e^{-\sigma_{tr}d_r}}{\sigma'_t d_r^3} + z_v(\sigma_{tr}d_v + 1) \frac{e^{-\sigma_{tr}d_v}}{\sigma'_t d_v^3} \right] \quad (11)$$

则多次散射 BSSRDF 形式如下:

$$S_d(p_i, \omega_i; p_o, \omega_o) = \frac{1}{\pi} F_t(\eta, \omega_i) R_d(\|p_i - p_o\|) F_t(\eta, \omega_o) \quad (12)$$

渲染算法

实际渲染中，在给定出射点的情况下需要采样附近的入射点，原文给出的方法是向光源发射阴影光线，得到同物体上的交点作为入射采样点，为提高采样效率，建议使用 $\sigma_{tr}e^{-\sigma_{tr}d}$ 作为采样的概率密度分布函数。

总结分析

该方法是第一个将扩散理论引入半透明材质渲染的工作，并提出解析式的BSSRDF表达式。缺点是其假定表面是半无限的、匀质的，并且该模型没有对入射方向、出射方向的影响考虑在内。

• 2002 年 Jensen 等提出的层次化积分加速模型 [8]

该工作在对偶极子模型基础上，提出两趟高效求解扩散近似的技术，并利用层次化积分技术加快计算，因此该方法也称为快速对偶极子模型。

根据对偶极子模型，某一点处辐射出射度表示为：

$$dM_o(x_o) = \frac{d\Phi_i(x_i)\alpha'}{4\pi} \left\{ C_1 \frac{e^{-\sigma_{tr}d_r}}{d_r^2} + C_2 \frac{e^{-\sigma_{tr}d_v}}{d_v^2} \right\} \quad (13)$$

$$\text{其中, } C_1 = z_r \left(\sigma_{tr} + \frac{1}{d_r} \right), C_2 = z_v \left(\sigma_{tr} + \frac{1}{d_v} \right)$$

两趟求解算法：

• 第一趟

基于泊松圆盘采样技术在物体表面获得均匀采样点（该采样技术介绍参见第一节），并且所需采样的数目和几何参数、光照、材质等均有关系，每个采样点处需记录其位置、面积（均匀采样的话该面积等于总面积除以采样数）、辐射照度估计值等。

• 第二趟

基于层次化的技术来快速求解辐射出射度：

数据结构

其数据结构使用八叉树，每个节点代表其所有子节点的照明贡献。中间节点需要存储该节点内总的辐射照度 E_v 、面积 A_v 、平均位置 p_v （按照每个子

节点的辐射照度作为加权重），而每个叶子节点存储不超过 8 个的采样点数据。

渲染算法

为计算某个点 x 处的出射辐射亮度 L_o ：

1. 从根部开始遍历八叉树；
2. 对每个节点：
 - a) 若其是叶子节点或者足够小¹则直接利用该节点数据求解，计算方法为：

$$M_{o,p}(x) = F_{dt}(x) \frac{dM_o(x)}{\alpha' d\Phi_i(x_i)} E_p A_p$$

其中， x 是待求点位置， p 是入射采样点位置， dM_o 见公式(13)；

- b) 否则，递归调用每个孩子节点进行求解（若 x 位于该节点代表区域的内部，也总是递归求解）；
3. 遍历完成后，得到总的 M_o ，需要将其转换为出射辐射亮度 L_o ， $L_o(x, \omega) = \frac{F_t(x, \omega) M_o(x)}{F_{dr}(x) \pi}$

总结分析

该方法在不引起明显误差的情况下，显著的提升了对偶极子模型的渲染效率，并且可以适用于所有基于辐射照度的半透明材质渲染算法。

下面介绍几种在对偶极子模型基础上做出改进和修正的模型。

• 2005 Jensen 等提出的多偶极子（multipole）模型 [9]

之前的对偶极子模型基于一个重要的假定：表面是半无限大的，而该假定在很多情况下并不成立，比如对于薄板或者多层的半透明材质就不满足该假定。

¹ “足够小”的度量标准：遍历节点内所有采样点，计算下面物理量的最大值： $\Delta w = \frac{A_v}{\|x - p_v\|^2}$ ，若该值大于某个给定的误差容限，则认为不够小，需要递归调用子节点计算；否则，认为其足够小，可以直接计算。

该文章提供多偶极子模型，来求解薄板情况下的扩散方程解，并结合薄板的结果和 Kubelka-Munk 理论来对多层材质建模。

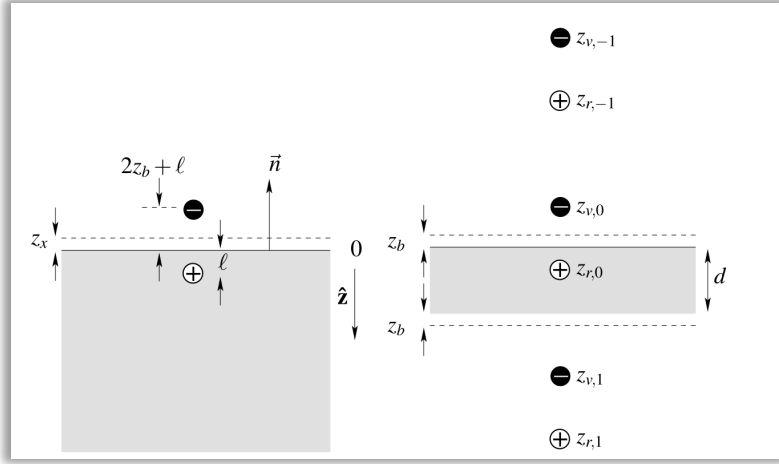


图1-9 半无限平面下的对偶极子（左） 薄板下的多偶极子（右）
该图片来自[9]插图

在经典对偶极子模型的基础上，修改边界条件来满足薄板假定（其厚度为 d ）¹：

$$\phi(r) + 2AD \frac{\partial \phi(r)}{\partial z} = 0 \text{ at } z = d$$

多偶极子模型通过 $2n+1$ 个对偶极子来代替原来的 1 个对偶极子使之满足新的边界条件（如图 1-8 所示），其中对偶极子的位置：

$$z_{r,i} = 2i(d + 2z_b) + l$$

$$z_{v,i} = 2i(d + 2z_b) - l - 2z_b, i = -n, \dots, n$$

$$\text{其中, } z_b = 2AD$$

此时，总漫反射反射率和透射率计算如下：

$$R_d(r) = \sum_{-n}^n \frac{\alpha' z_{r,i} (1 + \sigma_{tr} d_{r,i}) e^{-\sigma_{tr} d_{r,i}}}{4\pi d_{r,i}^3} - \frac{\alpha' z_{v,i} (1 + \sigma_{tr} d_{v,i}) e^{-\sigma_{tr} d_{v,i}}}{4\pi d_{v,i}^3}$$

$$T_d(r) = \sum_{-n}^n \frac{\alpha' (d - z_{r,i}) (1 + \sigma_{tr} d_{r,i}) e^{-\sigma_{tr} d_{r,i}}}{4\pi d_{r,i}^3} - \frac{\alpha' (d - z_{v,i}) (1 + \sigma_{tr} d_{v,i}) e^{-\sigma_{tr} d_{v,i}}}{4\pi d_{v,i}^3}$$

¹ 作为对比，原边界条件为： $\phi(r) - 2AD \frac{\partial \phi(r)}{\partial z} = 0 \text{ at } z = 0$ 。

而对于多层的材质，混合反射和折射即可：

$$P_d(r) = \frac{1}{2}(n_o \cdot n_i + 1)R_d(r) + \frac{1}{2}(1 - n_o \cdot n_i)T_d(r)$$

渲染方法

多偶极子可以采用光线跟踪采样的方法进行实现，也可以采用快速对偶极子模型中的层次化加速方法，唯一的区别在于需要对一系列的对偶极子的贡献进行求和后得到结果。

• 2011 年 d'Eon 及 Irving 等提出的量子对偶极子模型 [10]

经典对偶极子模型中，漫反射径向基函数 R_d 仅与积分通量率的梯度 $(\vec{n} \cdot$

$\vec{\nabla}\phi(p))$ 有关，与其本身 $\phi(p)$ 无关：公式表示为：

$$R_d(p) = C_\phi \phi(p) + C_E (-D(\vec{\nabla} \cdot \vec{n})\phi(p)) \text{ 而 } C_\phi = 0, C_E = 1$$

d'Eon 及 Irving 等从医学物理学、天体物理学等领域引入关于扩散理论的改进，包括更精确的反射参数 A、改进的扩散方程及更精确的辐射出射度的计算。需要指出的是，该工作同之前的经典对偶极子模型一样，仍然假定光线是垂直于表面入射来进行建模和计算的。

更精确的反射参数 A

$A(\eta) \approx \frac{1+2C_2(\eta)}{1-2C_1(\eta)}$ ，其中， C_1 C_2 采用有理近似得到，具体表达式参见附录。

改进的扩散方程

$$-D_G \nabla^2 \phi(p) + \sigma_a \phi(p) = \alpha' Q_0(p), \quad D_G = \frac{1}{3\sigma'_t} + \frac{\sigma_a}{3\sigma'^2_t}$$

在此基础上，推导辐射出射度：

$$\phi_G(p) = \frac{\alpha'}{4\pi D_G} \left(\frac{e^{-\sigma_{tr} d_r}}{d_r} - \frac{e^{-\sigma_{tr} d_v}}{d_v} \right)$$

$$r_{d,G}(p) = \frac{1}{4}(1 - 2C_1(\eta))\phi_G(p) + \frac{1}{2}(1 - 3C_2(\eta))(-D_G(\vec{\nabla} \cdot \vec{n})\phi_G(p))$$

进一步，该工作将对偶极子模型中的点光源扩展为光束光源，光束光源下的漫反射径向基函数：

$$R_{d,G}(p, \omega) = \int_0^{\infty} r_{d,G}(p, t) Q(t) dt, Q(t) = \alpha' \sigma'_t e^{-\sigma'_t t}$$

该公式没有闭式解，该工作中使用高斯函数和来近似求解，而 Habel 等人 [11] 指出该积分可以通过蒙特卡罗采样的方法来直接高效求解。

渲染中，可以采用蒙特卡罗采样的方式计算，但由于该方法考虑到了入射光方向的影响，无法直接采用快速对偶极子模型中的层次化加速方法。

- **2007 年 Donner, Jensen 等提出的光子扩散模型 [12]**

之前基于扩散方程近似的方法都只适用于匀质的、高吸光度的半透明材质，并且无法模拟诸如内部散射、体焦散、体阴影等效果。

该方法基于向半透明材质内部发射光子并基于这些光子的信息来层次化的计算积分，从而得到出射辐射亮度。

该方法引入四偶极子（quadpole）扩散近似来对直角边处的扩散行为建模、以及一个衰减核来对光源附近的多次散射进行准确近似。

该方法的渲染的方法和效率与蒙特卡罗光线跟踪实现的经典对偶极子模型一致，但可以更好的捕捉半透明全局照明效果，并可以处理斜入射、复杂几何体边界等情况。

- **2014 年 Frisvad, Hachisuka 等提出的方向对偶极子模型 [13]**

该方法基于 Menon 等人 [14] 提出的一种新的扩散方程的解来渲染半透明材质，该方法是第一个可以将入射光方向考虑在内的解析式模型。

该方法的主要思路是基于新的扩散方程解（对于某条光线，而非某个点），使用光线源代替经典对偶极子模型中的点光源，从而推导出漫反射径向基函数的表达式¹。

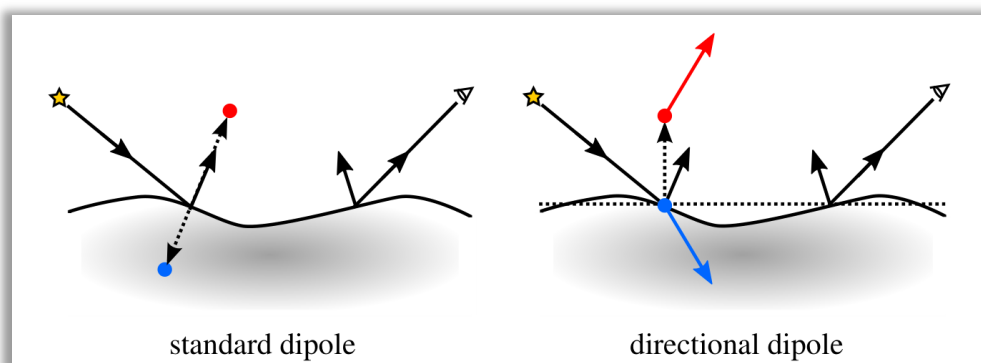


图1-10 经典对偶极子（左）和方向对偶极子（右）对比
该图片来自 [13] 插图

该方法可以使用蒙特卡罗采样的技术进行求解，和经典对偶极子模型的区别在于需要存储每个采样点的入射光线方向信息，论文中指出可以使用 $\exp(-\sigma_{tr}r)$ 作为俄罗斯轮盘技术的接受概率。该方法由于包含采样点的入射方向信息，无法直接使用层次化积分技术，而需要把辐射照度采样过程的每个采样都存储起来。

从 Jensen 等提出经典的对偶极子模型开始，后续发展出诸多的改进算法，由于此类方法实现容易、渲染效率高，并且可以实现较好的半透明材质效果，被广泛应用于渲染中。而接下来要介绍的方法则进一步放宽了对于材质、表面等的约束，

1.2.2.2 预计算方法

本小结将介绍两种典型的基于预计算来渲染半透明材质的方法，其中 Donner 等人的工作是本工作的基础。

- 2012 年 Yan 等提出的基于球面高斯近似的预计算方法

¹ 具体推导这里不再赘述，推导的过程和经典对偶极子模型、量子对偶极子模型等类似。

该工作使用球面高斯函数来表示环境光照，并基于 d'Eon 和 Irving 等人量子对偶极子模型中的漫反射辐射出射度表达式来得到解析式的渲染方法。

球面高斯

$G(i; i_c, \lambda) = \exp(\lambda(i \cdot i_c - 1))$, i_c 是球中心, λ 是其半径

球面高斯函数可以很好的近似环境光照，并且在乘法运算下封闭，其积分、卷积也存在解析表达，故被广泛用于图形学渲染中。

多次散射

$$L^D(x_o, o) = \int_{\Omega} G(i'; i_j', \lambda_j') \left(F \int_0^{\infty} Q(s) R_d(d) ds \right) di'$$

其中, $R_d(d) = (C_{\phi} \phi(d) + C_E (-D(n \cdot \nabla \phi)(d)))$

而基于扩散近似的积分通量 $\phi(d)$ 表达式如下：

$$\phi(d) = \frac{1}{4\pi D} \cdot \frac{e^{-\sqrt{\frac{\sigma_a}{D}}d}}{d}, d = \sqrt{s^2 + r^2 - 2sr(i' \cdot i_2)}$$

代入后可以将 L^D 分解为两项:

$$L^D(x_o, o) = FC_{\phi}L_{\phi} + FDC_EL_E$$

$$L_{\phi} = \int_{\Omega} \int_0^{\infty} G(i'; i_j', \lambda_j') Q(s) \phi(d) ds di'$$

$$L_E = \int_{\Omega} \int_0^{\infty} G(i'; i_j', \lambda_j') Q(s) (-n \cdot \nabla \phi)(d) ds di'$$

其求解基于以下方法：

- 使用球面高斯函数近似 $\phi(d)$;
- 基于指数函数拟合内部积分项，并通过制表的方法来存储拟合参数；

单次散射

$$L^{(1)}(x_o, o) = \sigma_s \int_0^{\infty} F_t E(s') \int_{\Omega} p G(i'; i_j', \lambda_j') E(s) V(i') di' ds'$$

其中 p 是相函数， G 代表折射后的入射光， E 代表衰减项， V 是可见性项，

s' 代表出射点 x_o 处折射光线路径的长度。将入射光和衰减项都使用球面高斯

函数近似，最终可以得到内部积分的解析表达，而外层的积分表示对折射方向进行采样来计算积分，容易在光线跟踪渲染器中实现。

- 2009 年 Donner, Jensen 等提出的经验 BSSRDF 模型 [16]

该方法基于向材质发射光子的模拟过程来采集原始数据，建立离散的表格数据，并基于二维投影中椭圆向圆转变的趋势提出基于椭圆坐标系的表达式进行拟合，最后将拟合数据制表用于渲染。

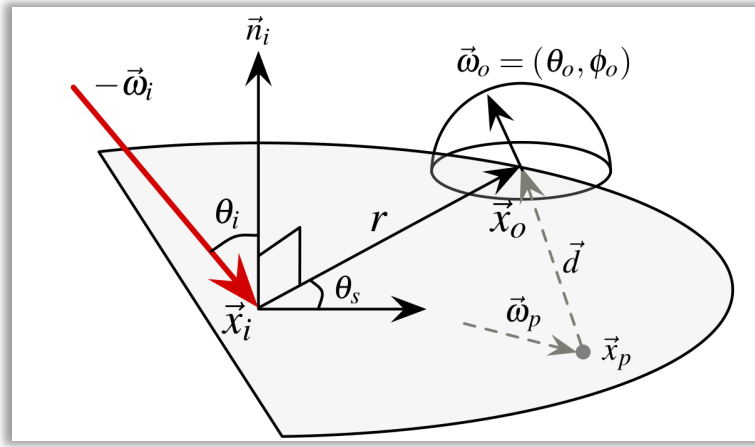


图1-11 BSSRDF 相关物理量，该图片来自 [16] 插图

首先将 BSSRDF 从 12 维缩减到 8 维：

$$S(\vec{x}_i, \vec{\omega}_i; \vec{x}_o, \vec{\omega}_o | \sigma_s, \sigma_a, g, \eta) \approx S(\theta_i, r, \theta_s, \theta_o, \phi_o | \alpha, g, \eta)$$

接下来通过模拟方法，采集数据，数据的维度为 $\{\theta_i, r, \theta_s, \alpha, g, \eta, \theta_o, \phi_o\}$ ：

$$\theta_i \in \{0, 15, 30, 45, 60, 70, 80, 88\}$$

$$r \in \{0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1, 2, 4, 8, 10\}$$

$$\theta_s \in \{0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180\}$$

$$\alpha \in \{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$$

$$g \in \{-0.9, -0.7, -0.5, -0.3, 0, 0.3, 0.5, 0.7, 0.9, 0.95, 0.99\}$$

$$\eta \in \{1, 1.1, 1.2, 1.3, 1.4\}$$

$\omega_o(\theta_o, \phi_o)$ 则采用 HEALPix 格式对球面进行 6208 个均匀采样

该文章提出的解析式表达：

$$H(\omega_o; \omega_{peak}, s_p, k_s | \Gamma) = k_s e^{-k_e \mu - k_c \chi} F_t(\omega_o, \eta) \quad (14)$$

其中， ω_o 代表出射方向， ω_{peak} 代表出射强度最大的方向， s_p 代表对称平面， k_s 是用于缩放整个分布（保证 ω_{peak} 处的值和模拟数据一致）， $\Gamma = \{a^+, a^-, b^+, b^-, k_e, k_c\}$ 表示其它的模型参数（控制形状、不对称程度、各向异性程度等）。

而 μ, χ 则基于椭圆坐标系，首先对于给定方向 θ_o, ϕ_o 获得其二维圆盘投影坐标：

$$\begin{aligned} x_d &= \frac{2\theta_o}{\pi} \cos\phi_o \\ y_d &= \frac{2\theta_o}{\pi} \sin\phi_o \end{aligned}$$

根据 (x_d, y_d) 可以求出其椭圆坐标系内的坐标 (μ, ν) ：

$$\frac{x_d}{a^\pm} + i \frac{y_d}{b^\pm} = \cosh(\mu + i\nu)$$

而 $\chi = \sqrt{x_d^2 + y_d^2}$ 。

该工作中使用迭代的优化算法对上述模型的参数进行拟合优化，得到每个输入变量 $(\theta_i, r, \theta_s, g, \eta, \alpha)$ 对应的 $\overrightarrow{\omega_{peak}} \overrightarrow{s_p} k_s \vec{\Gamma}$ 数据用于渲染。

渲染方法

该方法可以使用蒙特卡罗采样的方式进行计算，但无法直接使用层次化积分技术（需要进行一定的修改）。

具体的每个采样贡献计算如下：

- 根据 g, α, η 对原始表格数据进行插值；
- 计算模型所需的变量：

$$\begin{aligned} \vec{X} &= \vec{\omega}_i - (\vec{\omega}_i \cdot \vec{n}_i) \vec{n}_i, \quad \vec{Z} = \vec{X} \times \vec{n}_i \\ \theta_i &= \cos^{-1}(\vec{\omega}_i \cdot \vec{n}_i) \\ r &= (\sigma_a + \sigma_s) \|\vec{x}_o - \vec{x}_i\| \\ \theta_s &= \tan^{-1} \left(\frac{(\vec{x}_o - \vec{x}_i) \cdot \vec{Z}}{(\vec{x}_o - \vec{x}_i) \cdot \vec{X}} \right) \end{aligned} \tag{15}$$

- 根据 θ_i, r, θ_s 对数据进行进一步的插值,得到模型需要的参数

$$\omega_{wpeak}, s_p, k_s, \Gamma;$$

- 计算出射方向的球面坐标, 完成 BSSRDF 计算:

$$\begin{aligned}\vec{X}_h &= \vec{n}_o \times \vec{Z}, \quad \vec{Z}_h = \vec{X}_h \times \vec{n}_o \\ \theta_o &= \cos^{-1}(\omega_o \cdot \vec{n}_o) \\ \phi_o &= \tan^{-1} \left(\frac{(\vec{\omega}_o - (\vec{\omega}_o \cdot \vec{n}_o) \vec{n}_o) \cdot \vec{Z}_h}{(\vec{\omega}_o - (\vec{\omega}_o \cdot \vec{n}_o) \vec{n}_o) \cdot \vec{X}_h} \right)\end{aligned}\tag{16}$$

1.2.3 神经网络基本内容

神经网络是一种常用的机器学习方法, 在计算机视觉、自然语言处理等诸多领域取得了巨大的进展和成功。实验中使用了多层感知机网络来完成数据的拟合, 下面将简单介绍多层感知机网络的基本构造。

神经元

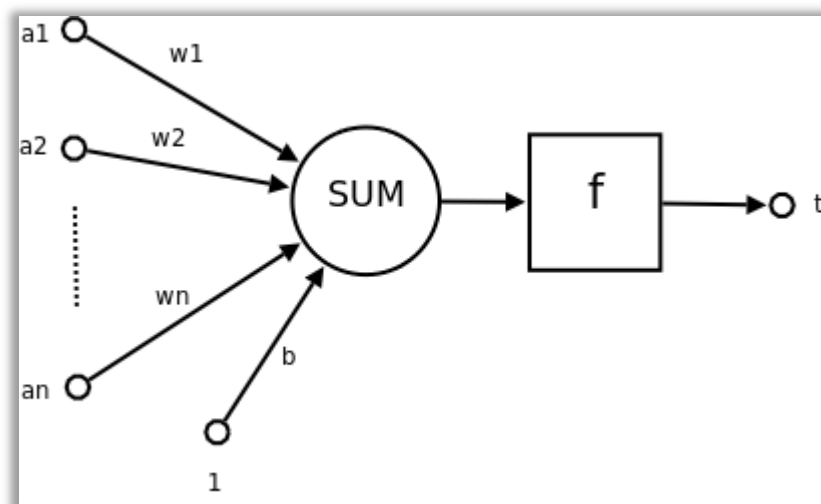


图1-12 神经元示意图

其中, $a_1 \sim a_n$ 为输入变量; $w_1 \sim w_n$ 为各个“突触”的权重; b 为该神经元的偏置; f 为激活函数 (非线性函数, 常用的包括 sigmoid 函数、tanh 函数、relu 函数等); t 为输出变量。

多层感知机网络

由三层以上（包含输入层、输出层以及隐层）的神经元连接而成的网状结构，其中每层的神经元都和前后层的每个神经元有连接关系（即每层都是全连接的）。隐层中每层神经元数目叫做该网络的宽度，网络的层数叫做其深度。

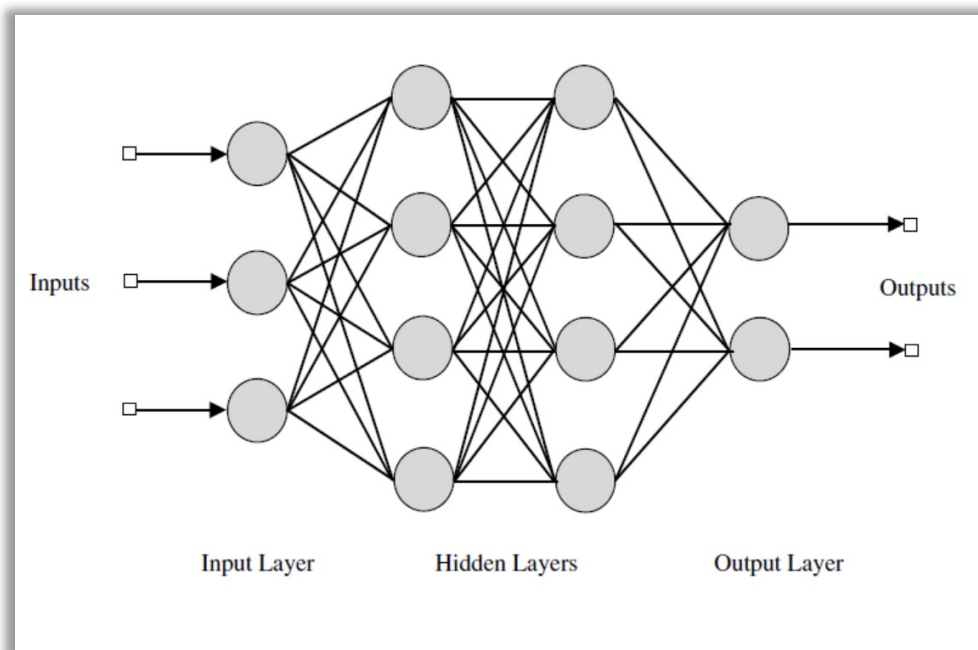


图1-13 多层感知机示意图

多层感知机网络通过随机梯度下降算法完成训练和学习过程，其中梯度的计算使用反向传播算法。梯度下降算法即在每轮计算后将权重按照梯度下降的方式进行调整，不断迭代该过程，直至最终损失函数小于给定值。

反向传播算法

记号说明： x 表示输入， a^l 代表第 l 层的输出值（应用激活函数后，激活前记作 z^l ）， w^l ， b^l 分别代表第 l 层的权重和偏置， \odot 代表阿达玛积（Hadamard product，即矩阵或者向量中对应元素相乘）。

1. 对输入 x ，计算输入层的输出 a^1 ;
2. 正向传播：对 $l = 2, 3, \dots, L$ ，计算 $z^l = w^l a^{l-1} + b^l$, $a^l = \sigma(z^l)$;
3. 计算输出层误差： $\delta^L = \nabla_a C \odot \sigma'(z^L)$;

4. 反向传播：对 $l = L - 1, L - 2, \dots, 2$ ，计算 $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$;
5. 输出梯度值： $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$, $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

2 正文

前一章介绍了真实感渲染的基本概念、半透明材质渲染的前人工作以及神经网络的简要内容。本章将从研究目的、内容、方法流程以及和前人工作的联系等方面入手介绍。简单而言，本文的工作基于 Donner, Jensen 等人模拟产生的数据以及模型，使用神经网络作为拟合工作，对该数据进行拟合，并将训练好的神经网络模型用于验证和渲染。

2.1 研究目的和动机

半透明材质渲染的方法中，基于扩散理论的方法大多拥有完全解析的表达式，渲染效率高，但它们大多需要进行一定的假设（对材质表面、材质属性、入射光分布、出射方向等），而这些假设在某些条件下很难成立，因此都拥有一定的局限性。

Donner 等人 2009 年的经验模型中，包含了入射方向、出射方向的因素，并且基于发射光子模拟的手段获得数据，结果较为准确。但该工作有两个缺陷：

3. 基于表格数据插值来计算，数据量大：

原始模拟数据总共有大约 70GB，而拟合后的模型参数数据大约有 300MB。

4. 更重要的一点在于，拟合数据的过程中，按照等高线进行逐层拟合，直至误差小于 1% 则结束迭代。该方法的问题在于虽然可以针对固定的 $\theta_i, \theta_s, r, g, \eta, \alpha$ ，将出射方向 θ_o, ϕ_o 的分布拟合的很好，但对于那些值较小（比如小于最大值 10% 的值）的出射方向，其相对误差会很大（原因在于：这些出射方向的值对最终的误差影响较小，故即使较大的相对误差也对最终的结果没有太大的影响）。

下面展示一个例子，分别选择不同出射方向绘制拟合模型、模拟数据随距离 r 变化的曲线，参数设置为 $\theta_i = 45, \theta_s = 150, g = 0, \eta = 1.3, \alpha = 0.99$:

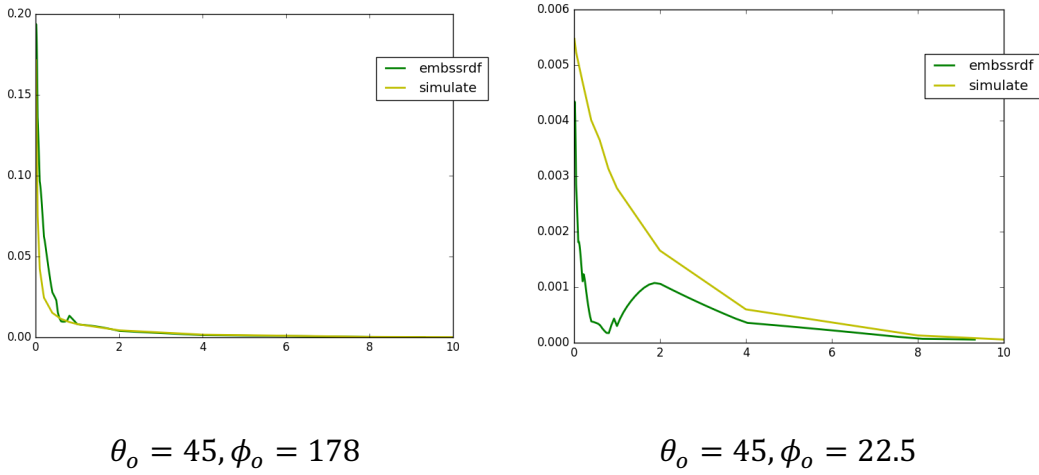


图2-1 不同出射方向 BSSRDF 随距离 r 变化曲线图

左侧的方向十分接近最大出射方向，拟合出的结果和模拟数据十分接近；而右侧的方向和最大出射方向距离较远，拟合出的结果比较差。

因此我们希望能够使用神经网络更好的拟合 Donner 等人提供的数据，以此来达到更加真实、准确的渲染效果；预期的模型是“端到端”的网络，即输入是 $\{\theta_i, \theta_s, r, g, \alpha, \eta, \theta_o, \phi_o\}$ 八维向量，输出是对应的 BSSRDF 值；这样训练得到的网络模型，可以很容易的集成到已有的半透明材质渲染实现中。

总之，本文提出的基于神经网络拟合数据的方法，首先可以压缩数据量，其次可以获得更好的拟合效果。

下面将介绍本文研究的主要内容。

2.2 研究内容

2.2.1 拟合策略

我们的第一项工作是对模拟数据进行拟合，其数据可以表示为 (\vec{x}, s) 的形式，其中 $\vec{x} = \{\theta_i, \theta_s, r, g, \alpha, \eta, \theta_o, \phi_o\}$ 代表 BSSRDF 的变量，而 s 代表 BSSRDF 的值。

由于神经网络模型表达力强且可以方便的利用 GPU 进行加速计算等优点，我们选择神经网络作为数据拟合的模型。

由于 BSSRDF 模拟数据十分庞大、输入-输出对应关系复杂，并且不同输出值之间的数量级差别很大（从 $1e-15$ 到 30），直接使用神经网络对其拟合难以收敛，会导致网络着重于去优化少数绝对值很大的数据点，而导致更多的点处拟合失败。

因此，我们参照 Donner 等人提出的解析式模型的形式（参见公式 14），提出我们的 BSSRDF 公式为： $S(p_i, \omega_i, p_o, \omega_o) = a \cdot e^b$ ，其中， a ， b 为待求系数。

神经网络无法直接对上式中的系数进行求解，我们提出的拟合策略为：

5. 对模拟数据的每个二维切片（二维切片指固定其它参数不变， θ_o, ϕ_o 改变而生成的子数据集），使用一个神经网络拟合其最大值的对数，记作 $\log(\max^i)$ ，其中 \max^i 代表第 i 个切片的最大值；
6. 同时，对模拟数据中每个数据，使用另一个神经网络预测其 BSSRDF 值和其所处二维切片最大值比值的对数，记作 $\log(s^j / \max^i)$ ，其中 s^j 指第 j 个样本数据， \max^i 代表第 i 个切片的最大值（假定第 j 个样本属于第 i 个切片）；

该拟合策略下，每个数据的输出值 $s^{j'} = a \exp(b)$ ，其中 a 是第一个神经网络的输出的指数（代表每个切片最大值的预测值）， b 是第二个神经网络的输出（代表每个数据和最大值的比例关系）。

而作为对比，Donner 等人提出的解析式 BSSRDF 公式中，比例因子 k_s 的含义也是代表每个二维切片的最大值，而指数因子 $-k_e \mu - k_c \chi$ 中 μ 代表其所处等高线的大小（参见图片 2-2 所示）， χ 代表距离最大值点的距离，因此该指数因

子也可粗略的理解为每个数据点和最大值的比例关系。和该模型不同，利用神经网络直接进行拟合，无需通过建模成椭圆坐标系来求解，有更大的自由度和准确性。

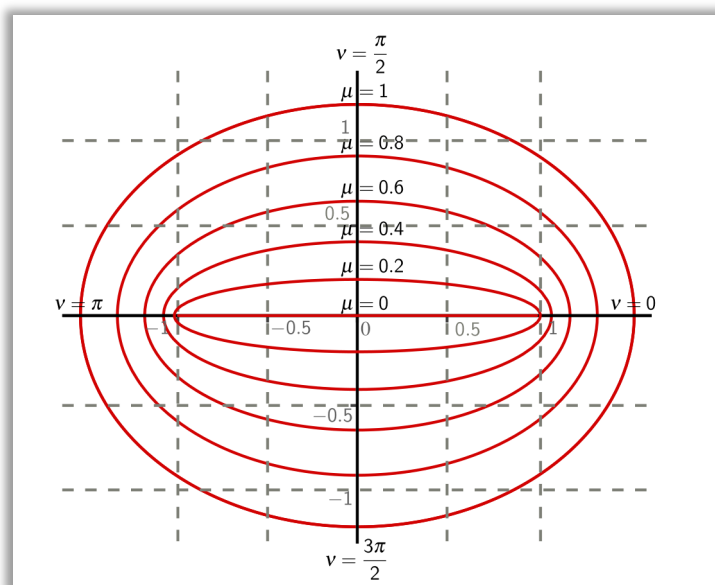


图2-2 椭圆坐标系示意图，该图片来自[16]插图

2.2.2 神经网络训练

在确定好基本的拟合策略后，需要通过训练神经网络来完成拟合工作，而训练过程中需要对数据进行预处理以及优化网络的结构来达到更好的拟合效果。

2.2.2.1 数据预处理

我们使用的原始数据来自 Donner 等人的附加材料¹，包括模拟数据以及拟合数据，我们这里仅使用模拟数据用于神经网络的训练（拟合数据仅用于实现 Donner 等人的算法用于对比使用），下面简单介绍一下基本的预处理过程：

¹ 数据在 ACM 网站上可以直接下载，网址为 <http://dl.acm.org/citation.cfm?id=1531336>

由于原始数据是二进制文件（使用 Healpix 格式¹存储），需要进行特别的读取操作，我们将数据首先转换成文本格式方便后续的处理。

其次，我们将数据整理成{X;Y}的形式，其中每一条记录都包含输入X及输出Y。

最后，根据前面介绍的拟合策略，我们还需要进一步的处理：

- 训练二维切面最大值的网络

将每个二维切面内的数据全部聚合成一条记录，去掉输入中的

θ_o, ϕ_o 变量，输出则是这些数据输出的最大值的对数。

- 训练二维切面比值的网络

修改每条记录的输出值，即每条记录除以它所处的二维切面的最大值，再取其对数即可。

至此，我们已经可以得到用于训练两个网络的全部数据。

进一步观察发现，相当一定比例的数据输出值均非常小(例如 $1e-10$ 以下)，这些值在渲染中难以有明显的贡献，但由于其取对数后绝对值很大，会对神经网络的损失值造成很大的影响，因此我们在预处理中将极小的数据过滤掉，采用的方法是将训练数据分成两部分，一部分是输出值较大的，另一部分则是输出值很小的。第一部分的数据是我们要拟合的重点，另一部分我们可以直接选择丢弃或者使用另外的网络去拟合它，然后通过一个二分类的过程结合两套网络。

2.2.2.2 神经网络结构及超参数的选择

我们选择多层感知机模型进行拟合，为达到更好的效果，激活函数、学习方法等的选择以及网络的宽度、深度、学习速率等超参数都需要在实验验证中去确定。

¹ Healpix 是 Hierarchical Equal Area isoLatitude Pixelization 的缩写，用来产生球面的等面积均匀采样，具体内容见 <http://healpix.jpl.nasa.gov/>

实验设计的方法就是为每个超参数设置一个梯度，在每个取值上进行一次小规模训练尝试，迭代一定轮数后记录其损失值，损失最小的取值作为最终该超参数的取值。比如对于激活函数的选择，我们首先对原始数据集进行一定的下采样生成子数据集，分别训练两个网络，其中一个使用 Tanh 函数作为激活函数，而另外一个使用 Relu 函数作为激活函数，迭代一定轮数（例如 100 轮）后观察二者的损失变化，选择更好的那个作为之后拟合的激活函数。

除了损失值外，还考察了在一些测试样本上模型预测的好坏来帮助我们做进一步的选择。实验中，使用模型在某些样本点处生成二维 BSSRDF 切片的投影图片来作为其好坏的度量，这样做比单纯的损失值更加直观，且更容易体现出模型预测的好坏。

2.2.3 半透明材质渲染的实现

在神经网络拟合数据的基础上，我们需要将得到的模型应用到渲染当中去。第一章中已经介绍过，半透明材质渲染算法大多基于蒙特卡罗光线跟踪或者层次化加速技术来实现，我们的方法也不例外。下面具体介绍这两种渲染方式的原理和方法。

2.2.3.1 基于采样的方法

半透明材质渲染方法容易集成到基于蒙特卡罗光线跟踪的程序当中，基本思路是利用蒙特卡罗采样技术来求解渲染方程，参见公式(4)：

$$L_o(p_o, \omega_o) = \int_A \int_{\Omega} S(p_o, \omega_o, p_i, \omega_i) L_i(p_i, \omega_i) |\cos \theta_i| d\omega_i dA$$

其对应的蒙特卡罗估计量为：

$$\bar{L}_o = \sum_{k=1}^N f^k = \sum_{k=1}^N \frac{1}{N} \frac{S(p_o^k, \omega_o^k, p_i^k, \omega_i^k) L_i(p_i^k, \omega_i^k) |\cos \theta_i^k|}{p(\omega_i^k) p(p_i^k)}$$

其中， $p(\omega_i^k)$ 代表采样某个入射方向的概率， $p(p_i^k)$ 代表采样某个入射点的概率。

实现中只需要在每个出射点处，得到若干个入射点的采样，然后计算贡献值即可得到每个出射点的出射辐射亮度值的估计量。

我们的渲染方法是两趟的算法，第一趟用于表面均匀采样，第二趟用于渲染时计算出射辐射亮度。采用表面均匀采样可以简化实现，并获得同样的渲染效果。

预处理

第一趟是预处理步骤，负责获得物体表面（总面积为 A ）的均匀采样点（共 N_{total} 个），并记录每个采样点的入射辐射亮度值 L_i 、入射方向 ω_i 、表面法向量 n_i 、位置坐标 p_i 、采样的概率密度函数 $p(\omega_i)$ 、采样面积 A_p 等。

其中位置、法向量信息通过均匀采样例程已经可以获得，而入射方向和入射辐射亮度则需要在每个点处向光源发射采样光线来获得，该过程在光线跟踪中容易实现故不再赘述。我们在每个点处进行 32 个入射方向及辐射亮度的采样。另外由于是均匀采样，每个采样点的面积相等。

渲染

第二趟是渲染步骤，利用第一趟得到的入射点信息来计算出射辐射亮度的蒙特卡罗估计量。

具体来说，在每个出射点处，从入射点列表中随机选择 N 个作为入射点，然后应用俄罗斯轮盘技术决定每个入射点是否使用（接收概率为 $\exp(-d\sigma_{\text{tr}})$ ¹。

¹ 接受概率是一个浮点数，而 σ_{tr} 是三维向量，故渲染中一般取其三个分量中最小的值计算接受概率。

而对于每个出射点、入射点对 (p_i, p_o) 而言，按照 Donner 等人论文中的方式计算参数 $\{\theta_i, \theta_s, r, g, \alpha, \eta, \theta_o, \phi_o\}$ （参见公式 15、16），然后使用 BSSRDF 模型¹ 计算对应的 BSSRDF 值 S^k ，

$$\text{记 } L_{\cos}^k = \frac{L_i \cdot |\cos\theta|}{p(\omega_i)}, \text{ 有 } p(p_i^k) = \frac{1}{A_p} \cdot \frac{A_p}{A} \cdot \frac{N}{N_{total}} = \frac{N}{A \cdot N_{total}},$$

则对应的蒙特卡罗估计量的贡献值如下：

$$f^k = \frac{1}{N} \frac{S^k L_{\cos}^k}{p(p_i^k)} = \frac{S^k L_{\cos}^k A \cdot N_{total}}{N \cdot N} = \frac{S^k L_{\cos}^k A_p N_{total}}{N}$$

需要注意的是，由于使用了俄罗斯轮盘技术来加速，需要修改 f^k 如下：

$$f^k = \frac{S^k L_{\cos}^k A_p N_{total}}{N p_{accpet}} = \frac{S^k L_{\cos}^k A_p N_{total}}{N e^{-d\sigma_{tr}}}$$

2.2.3.2 基于层次化加速的方法

原始的层次化加速方法参见第一章中 Jensen 等人 2002 年提出的快速对偶极子模型中的介绍，而该实现中预采样的过程中记录了每个采样点的辐射照度，而辐射照度是对入射方向的积分，因此并不包含入射方向信息，这就导致该方法无法直接应用到考虑入射光方向因素的模型当中，比如方向对偶极子模型、embssrdf 模型以及我们的模型。

而通过对原方法进行一定改造就可以将入射光方向因素考虑在内，下面具体介绍其内容 [13]。

本方法同样是两趟算法，第一趟用于获得表面均匀采样点并建立八叉树，而第二趟通过遍历八叉树来计算射出辐射亮度。

预采样

¹ 比如使用神经网络进行预测或者 Donner 等人提出的解析式模型对 BSSRDF 进行求解，另外只使用其中的部分参数也可以直接使用对偶极子模型、方向对偶极子等模型进行求解。

每个采样需要保存的信息同之前基于光线跟踪的方法。之后利用这些采样点信息建立八叉树，和之前快速对偶极子模型的区别在于不再存储辐射照度采样，而是将辐射照度采样过程中的入射方向列表、入射辐射亮度列表均存储起来。（为方便采样点的聚集，对每个采样点使用相同的一组入射方向）；

渲染

遍历八叉树的方法和快速对偶极子模型相同，区别在于计算每个节点的BSSRDF值贡献时，需要对所有的入射方向进行遍历来计算。

原快速对偶极子模型将方向、位置采样点均聚集为一个节点，而本方法则仅将位置采样点聚集，方向需要通过遍历的方法考虑，因此速度会比原方法慢，但仍明显快于直接遍历采样点进行计算。

由于该方法带来的效率提升有限，且采样率不够会呈现明显的条带状瑕疵，因此我们实验中，仅针对方向光源的情况对所有渲染算法实现了层次化加速技术的版本，而对于其它光源条件下，则使用基于光线跟踪采样的方法进行渲染（该情况下，采样不足会体现为噪音）。

以上就是我们工作的原理方法，主要介绍了拟合的整体思路和策略以及渲染算法的实现方法。下面将详细介绍一下具体的流程步骤。

2.3 实验方法和流程

2.3.1 拟合数据

2.3.1.1 实验环境

软件环境：

我们采用了 Python 语言的 Keras 库 [17] 来搭建神经网络，并使用 Tensorflow [18] 作为其后端。

拟合实验在 Ubuntu 14.04 操作系统下进行。

硬件环境：

训练过程在 GPU（图形处理单元）上进行，实验中用到两块 NVIDIA GTX1080 来对训练加速。

2.3.1.2 流程

- 数据预处理

预处理的具体步骤参见前一节的介绍。这里作一些补充说明：

- 原始数据（文本格式）大约为 35G，共 $9.38e8$ 条记录¹；
- 数据过滤中，我们将取值大于 $1e-6$ 的作为较大的部分进行训练，其他部分则舍弃；
- 对于训练最大值而言，数据量一共为 151008 条数据，虽然可以直接用于训练，实验中仍然采用了批量的方式输入数据，具体参见下面介绍；
- 对于训练比值而言，数据量很大，无法直接放入内存（或 GPU 显存），需要通过 Keras 提供的批量处理接口²。为方便文件读取效率，我们将原数据文件分解成若干个文本文件，只考虑较大部分（即将输出小于 $1e-6$ 的数据过滤）的话，每个文件包含 48423 个样本数据，共有 11413 个文件。

- 网络结构和超参数的确定

¹ 这里指对于某个特定的折射率 η 而言，对整个数据集的话需要再乘 5。

² 即 Keras 库中 `model.fit_generator` 方法，具体参见 <https://keras.io/models/sequential/>

要确定参数和结构主要包括：激活函数、学习方法、网络宽度、深度以及学习速率、衰减率等。

通过实验对比，最终我们确定的网络结构为：

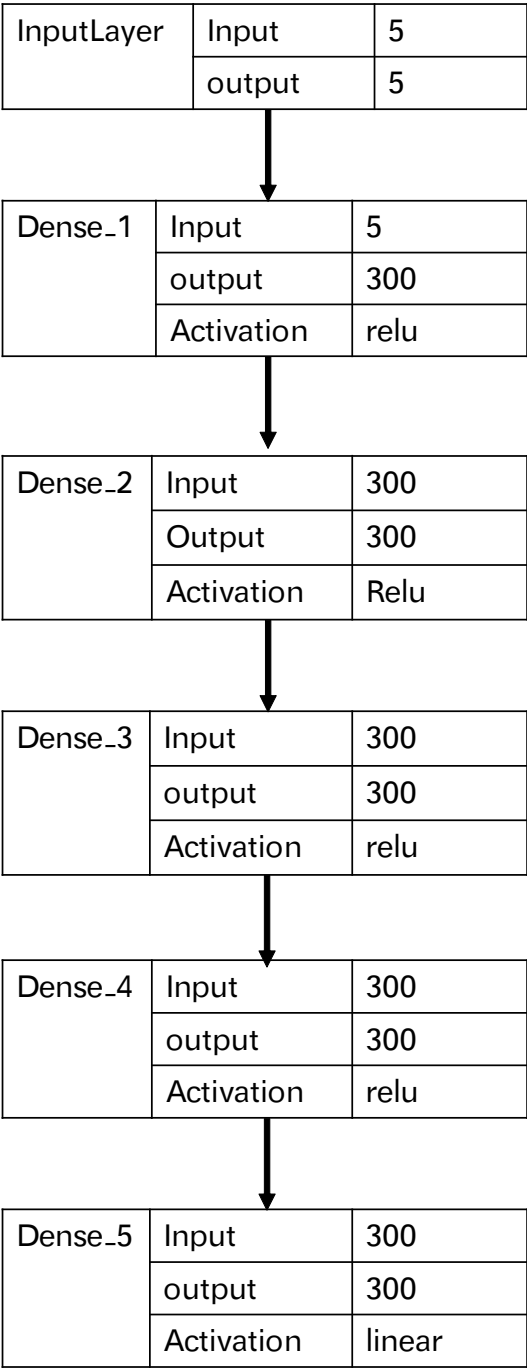


图2-3 训练二维切片最大值网络结构

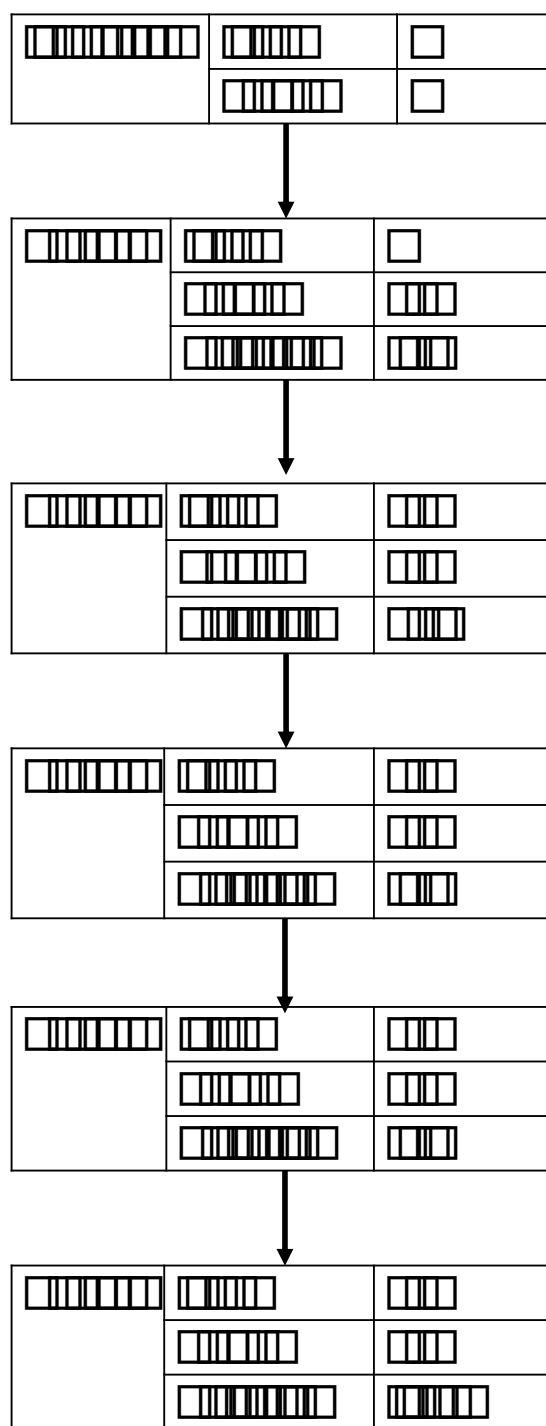


图2-4 训练二维切片比值的网络结构

而学习率初始化为 $1e-4 \sim 1e-5$ 之间，衰减率在 $1e-5 \sim 1e-6$ 范围内，需要注意的是，学习率在学习过程中可以通过观察损失曲线来人为进行调整和修正。

- 训练网络并测试

在确定网络结构和超参数后，直接在 Keras 中建立网络，并设置相应的学习算法及参数，然后给定特定的数据产生器，即可开始训练。

训练中，除打印每趟的损失值外，我们还使用模型预测某个输入下的二维切片投影图来进一步帮助我们判断收敛情况。

2.3.1.3 其它问题

训练中除之前介绍的通用的超参数选择外，由于采用批处理的方法进行训练，批量大小(batch size) 也需要确定。除考虑批量大小对准确度的影响外（批量越小越精确，但训练抖动会更严重），还需要考虑其对训练效率的影响（批量越大，一般训练会越快）。

另外，由于数据量过大，即使使用批量处理，训练一趟的时间也在 20 分钟以上，因此我们使用了下采样的方式来进一步加快训练过程，即随机选择每个文件中 10% 的数据用于训练。

2.3.2 渲染实现

之前已经介绍过基于光线跟踪以及层次化加速技术实现半透明材质的原理方法，这里将列举实现的流程步骤以及一些实现细节补充。

2.3.2.1 实验环境

我们将算法实现于开源渲染器 Mitsuba [19] 中，以扩展其 subsurface 类的形式实现我们的算法，而光线跟踪的基础设施、物体表面均匀点采样以及层次化技术中用到的八叉树结构等可以直接使用 Mitsuba 中的实现。

Mitsuba 是跨平台的渲染器，实验在 Windows 10 及 Ubuntu16.04 下进行。

由于神经网络前向传播中需要进行矩阵乘法的运算，因此我使用了开源线性代数库 Eigen¹来实现。

2.3.2.2 流程

基于光线跟踪采样的方法

我们将 embssrdf 模型、对偶极子模型、方向对偶极子模型及我们的模型均实现于该框架中，依靠一个整型变量来决定使用哪种渲染策略。下面是具体实现步骤：

- 预处理中，首先生成物体表面的均匀采样点，在每个采样点处获得 32 个入射光采样（包括入射光方向和辐射亮度²）；之后根据渲染方法的不同，进行相应的初始化： embssrdf 模型需要读取拟合数据并根据 $g \propto \eta$ 对表格数据进行插值；我们的模型则需要从文件中加载两个神经网络；其它两种方法不需要其它的预处理步骤。
- 渲染中，首先得到计算 BSSRDF 需要的输入变量（比如 θ_i, θ_s, r 等），然后随机选择均匀采样点中的若干个，借助俄罗斯轮盘技术来过滤较远的采样点，然后计算每个被接受的采样点的贡献值。

基于层次化加速的方法

如前面所述，我们仅针对方向光源条件下，实现这几种算法。在该简化下，入射光方向对每个采样点都是固定量，因此我们只需要简单修改一下原始的快速对偶极子模型即可：首先每个采样点需要额外保存法向量信息，其次需要对每个节点内的法向量进行平均来得到节点的法向量。

下面介绍其流程：

- 预处理中，同样是首先生成均匀采样点，然后建立八叉树结构，最后根据渲染方法不同进行特定的初始化操作（同之前的方法，不再赘述）；

¹ Eigen 的信息介绍参照 eigen.tuxfamily.org/

² 为方便起见，这里保存的入射辐射亮度的值实际上是其乘以 cosine 项后的结果。

- 渲染中，通过遍历八叉树结构来得到出射点的辐射亮度，而计算每一个节点的贡献值时：首先根据几何信息计算输入变量，然后利用该信息计算 BSSRDF 值，进而得到其贡献值。

该方法中，我们同样实现了对偶极子模型、方向对偶极子模型、embssrdf 模型以及我们的模型，它们的区别仅存在于计算 BSSRDF 的方法不同，所使用的渲染流程是完全一致的。

3 结论和讨论

前一章从我们研究动机出发，介绍了拟合数据的策略方法和两种渲染策略的原理和过程，本章将展示实验得到的结果，包括拟合结果分析及渲染结构展示两部分，来证明我们模型的准确性。

3.1 拟合结果

3.1.1 神经网络训练误差

预测二维切片最大值的网络最终训练误差为 0.00090，而预测比值的网络最终训练误差为 0.030；此外，针对 $g=0$ （代表各向同性）的情况，预测比值的网络训练误差可以降到 0.00166 左右。

由于训练数据为离散的表格数据，并且不适合直接插值来生成新数据，因此没有测试集可以使用（所有的数据都作为训练数据使用，以提高结果准确性）。因此，我们将从其它方面入手，验证该模型的准确性。

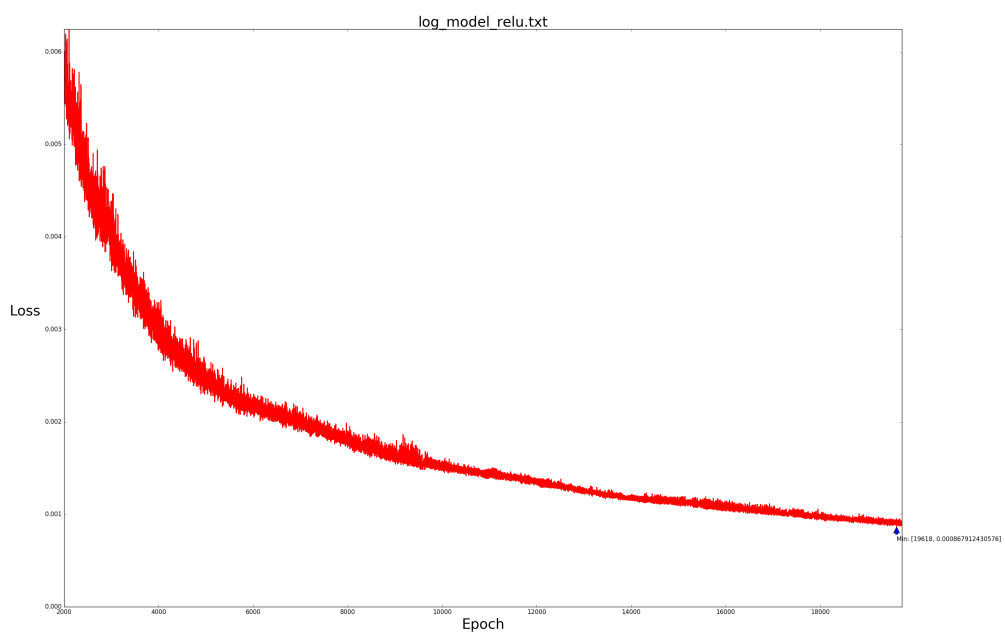
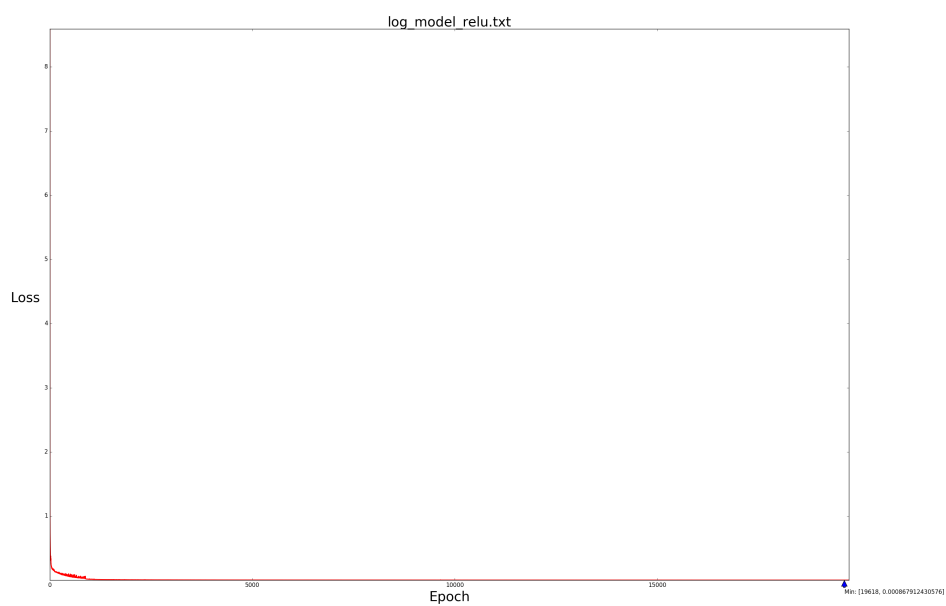


图3-1 图 a 展示了训练二维切片最大值过程中损失值下降曲线
图 b 展示了迭代 2000 趟之后的局部。

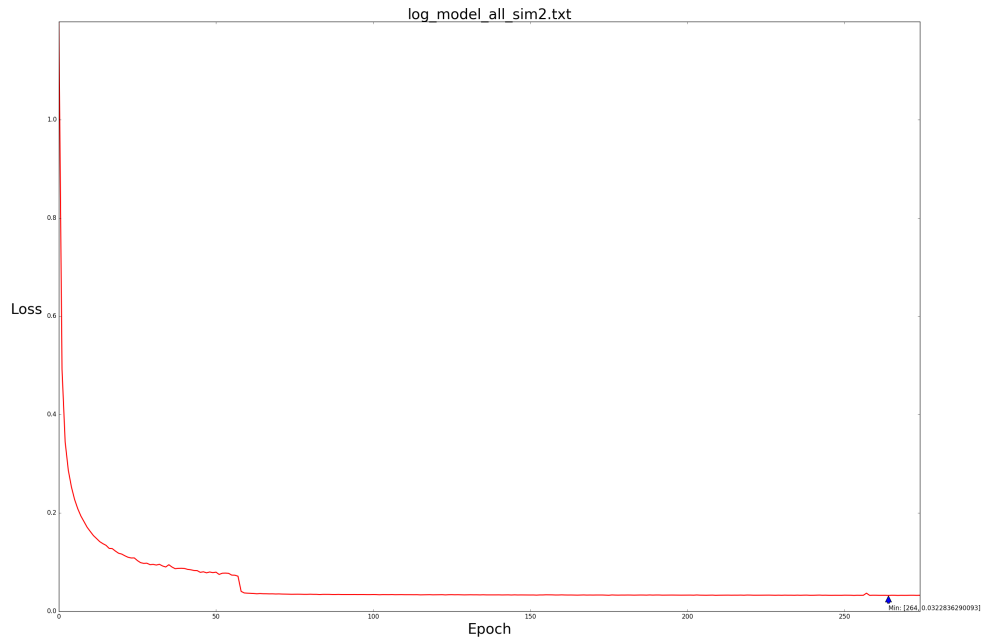
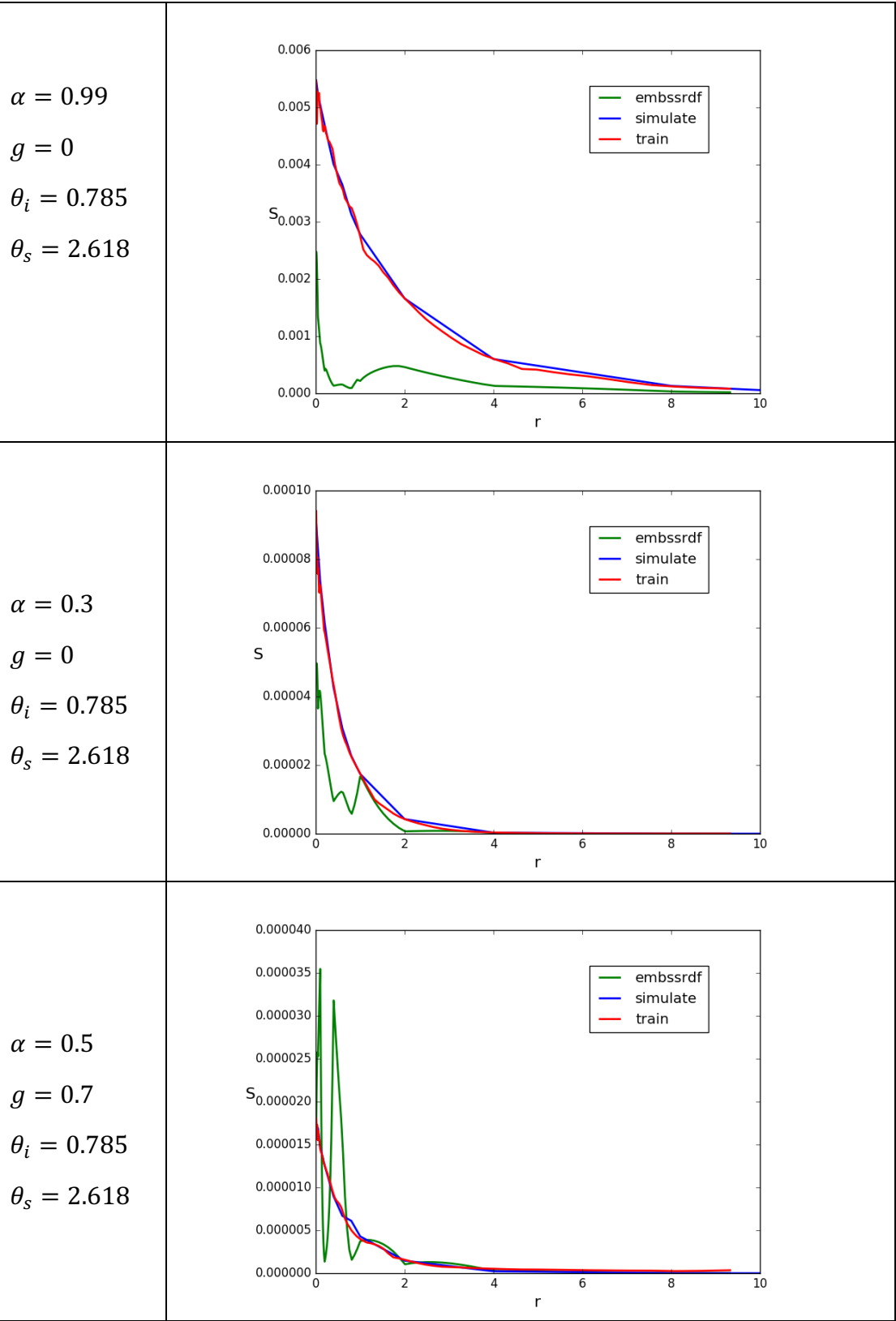


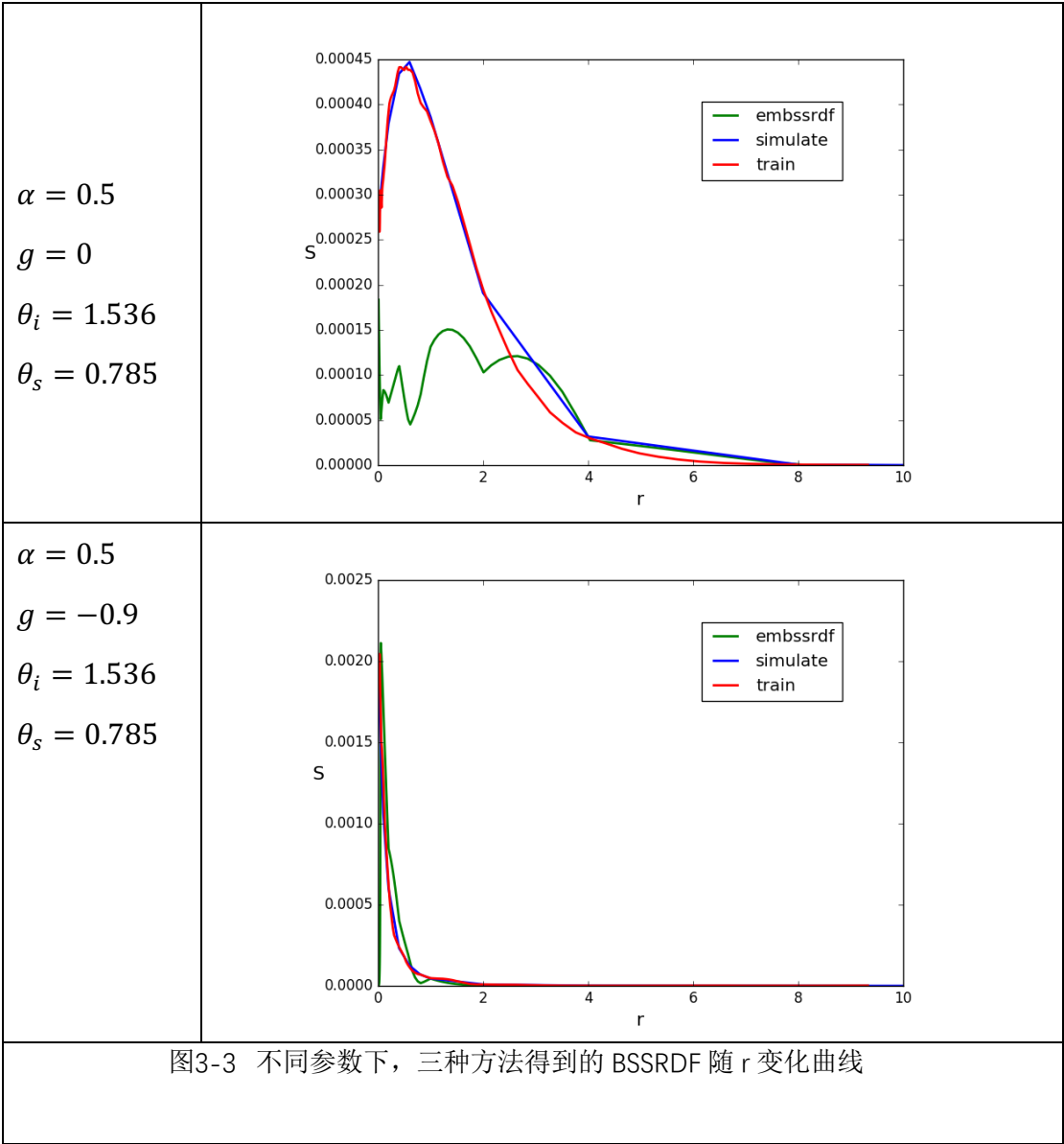
图3-2 训练二维切片比值的损失下降曲线

3.1.2 BSSRDF 随距离 r 变化曲线

这里将展示不同参数设置下，BSSRDF 值随着距离 r （入射点和出射点间距离）变化而变化的曲线图。其中，出射方向的因素将在下一小节专门考察，因此这里固定 $\vec{\omega} = (\theta_o, \phi_o) = (0.785, 0.393)$; η 对结果影响比较平缓，因此固定为 $\eta = 1.3$; 换言之，变化的参数包括 $\theta_i, \theta_s, \alpha, g, :$

图 1 和图 2 对比展示了吸光度的影响（低吸光度的情形是传统基于扩散理论无法捕捉的），图 3 和图 5 则展示了各向异性因子的影响，图 4 展示了入射、出射点几何关系 (θ_i, θ_s) 的影响。





可以看出，在各种参数设置下，我们的模型都保持了和模拟数据变化趋势的一致性并且取值也十分接近，而作为对比，embssrdf 模型则在某些情况下会出现趋势不一致（比如震荡），并且取值上也有较大的误差。

需要注意的是，存在少数情况下，训练模型会出现比较大的偏差，一般出现在整条曲线取值很小且很平缓，比如下图所示的情形中模拟数据中最大值也只有 $2e-6$ ，难以准确预测。

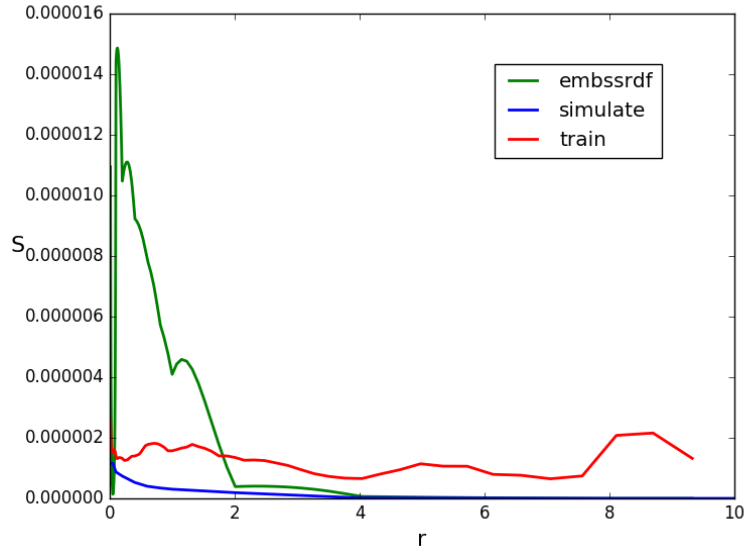


图3-4 参数设置为: $\alpha = 0.5$ $g = 0.9$ $\theta_i = 0.785$ $\theta_s = 2.618$

3.1.3 二维投影图拟合结果¹

参考 Donner 等人的工作，这里将展示可视化 BSSRDF 二维切片投影到单位圆盘上的结果。图 x-x 展示了不同参数设置下的结果对比。最左侧列是参数设置说明²，第二列代表模拟数据的结果，第三列代表 Donner 等人 embssrdf 模型的结果，第四列是我们提出的基于神经网络的模型的结果，最右侧列代表颜色映射。

具体来说，该示意图生成的时候需要将半球面映射到圆盘上；对单位球面上一点 $\vec{\omega}_o = (\theta_o, \phi_o)$ ，将其映射到单位圆盘上 (x_d, y_d) ，该映射关系如下：

$$x_d = \frac{2\theta_o}{\pi} \cos \phi_o, y_d = \frac{2\theta_o}{\pi} \sin \phi_o$$

对于模拟数据结果图的生成，需要注意的是，同 Donner 等人利用光子追踪模拟产生结果不同，我们使用离散的表格数据进行简单的插值后生成示意图，由于该表格将半球面分成 6208 块，因此分辨率有限。

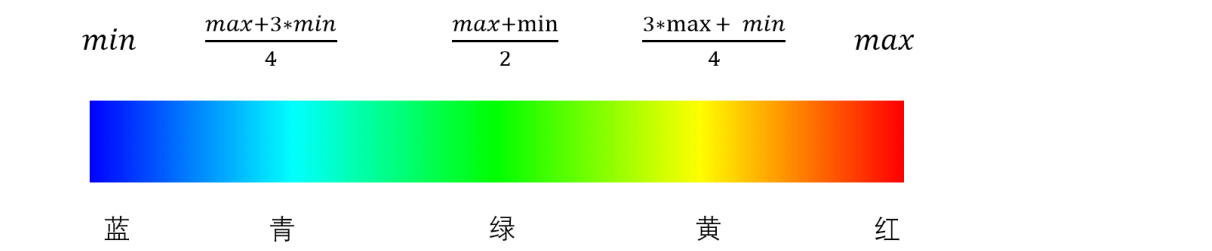
¹ 该结果的生成可以参考附录中的代码。

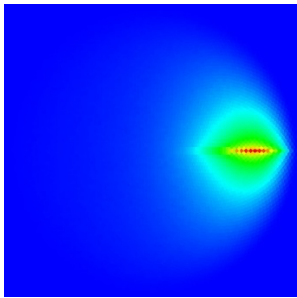
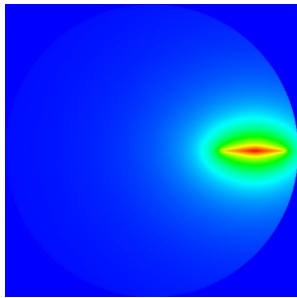
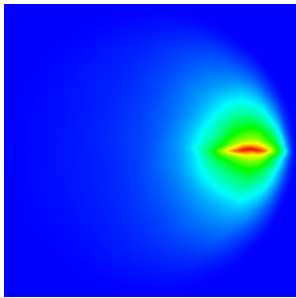

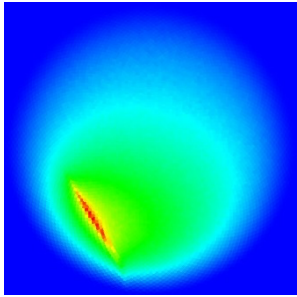
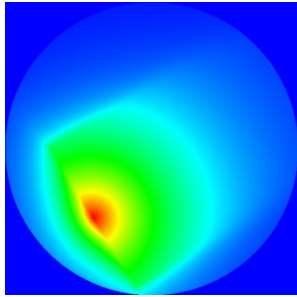
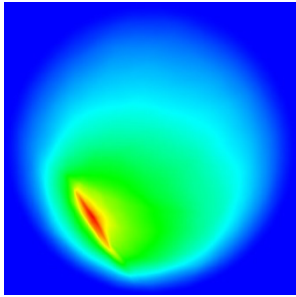

² 为方便起见，下面的示意图的参数中折射率 η 均为 1.3，不再一一罗列。

对于 embssrdf 模型结果图的生成，采用和渲染相同的方法，即在表格中插值确定参数后，使用解析式公式生成。

对于神经网络模型结果图的生成，这里可以忽略预测最大值的网络，而只使用预测比值的网络进行绘制（因为绘制图片的时候会进行归一化）。

颜色映射中，蓝色代表最小值零，而红色代表最大值 1，具体映射过程中，以蓝、青、绿、黄、红¹为边界点分成 4 个区间，在区间内部使用线性插值。



参数	模拟数据	原模型	神经网络模型	颜色
$\theta_i = 30^\circ$ $\theta_s = 0^\circ$ $r = 4$ $\alpha = 0.6$ $g = 0$				
$\theta_i = 30^\circ$ $\theta_s = 0^\circ$ $r = 4$ $\alpha = 0.6$ $g = 0$				

¹ 这几种颜色的 RGB 值为：蓝 (0, 0, 1)，青 (0, 1, 1)，绿 (0, 1, 0)，黄 (1, 1, 0)，红 (1, 0, 0)

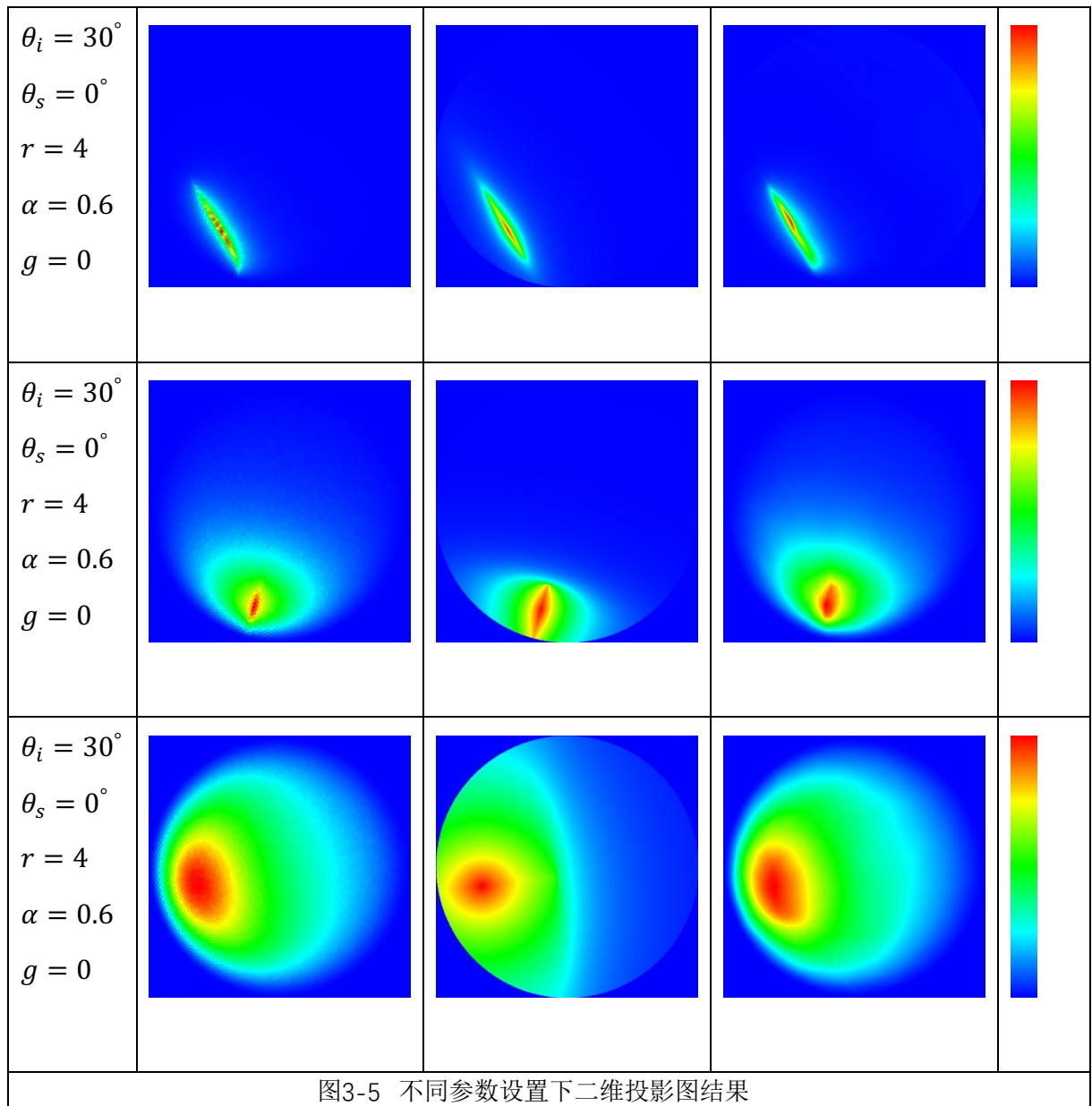


图3-5 不同参数设置下二维投影图结果

从上面的结果中可以看出，对于各种参数设置，我们的基于神经网络的模型都可以准确的捕捉该二维切片上的值，并且比原 embssrdf 模型要更加接近模拟数据模型。

下面列出每种情况下 embssrdf 模型和我们的模型的均方误差值（RMS）¹，其中每个数据都保留 4 位有效数字，计算方法为：

¹ 表格中的数据和 Donner 等人论文中的并不一致，原因在于用于参考的模拟结果，我们是使用离散数据插值得到的，因此误差都会相应的大一些，但不影响其比较。

$$RMS = \sqrt{\frac{\sum_{i=0}^n (x_i - x_i')^2}{n}}$$

另外，下面的数据表示了 BSSRDF 的 RMS 值，其单位是 $m^{-2}sr^{-1}$.

表3-1 RMS 结果对比		
编号	Embssrdf 模型	我们的模型
1	2.243e-005	4.847e-006
2	0.002060	2.941e-4
3	7.322e-6	4.491e-6
4	9.592e-6	1.296e-6
5	0.0001330	1.129e-5

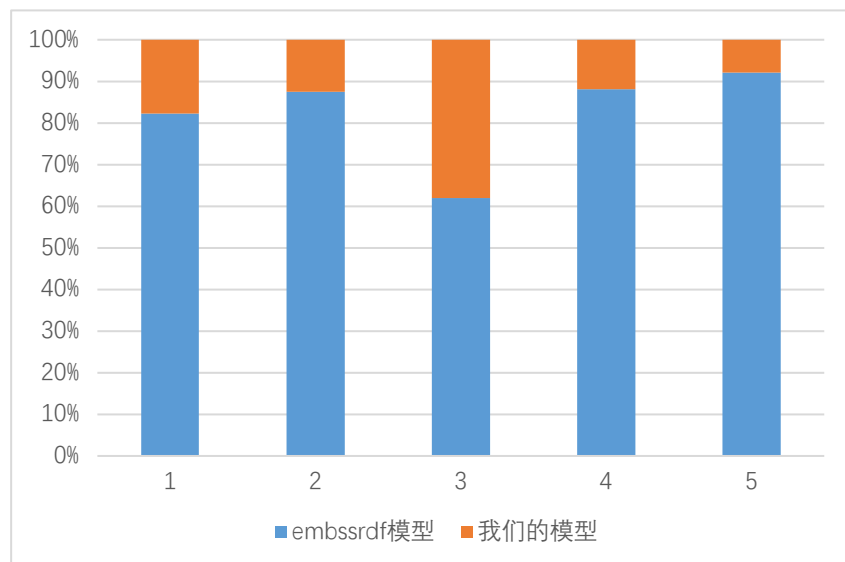


图3-6 RMS 百分比示意图

从上面的 RMS 结果也很容易发现，在各种参数设置下，我们的模型均拥有更小的 RMS 值，即有更好的拟合效果。

3.2 渲染结果

本节将展示在实际渲染中，该模型和其它模型的对比。

3.2.1 简单场景渲染结果

本部分通过对简单形体（球体、立方体）进行半透明材质的渲染来展示不同渲染方法的结果。

3.2.1.1 场景设置

具体的场景相机、光源参数可以参考附录中的 Mitsuba 场景格式文件。这里采用阴影的方式直观展示。

下图中展示了球体和立方体的场景设置，注意在真正渲染的时候会变化相机使物体在成像平面中央。

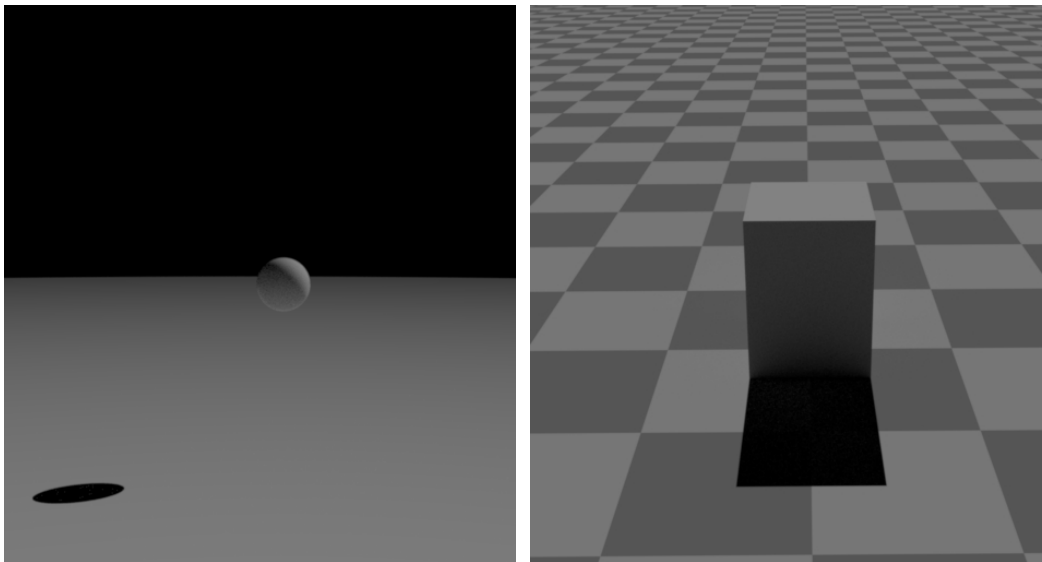


图3-7 场景、光照示意图

3.2.1.2 球体、立方体结果展示

下面的结果中，单次散射结果使用光线跟踪算法直接得到，再累加到半透明材质渲染的多次散射结果中。

球体和立方体都是几何上非常简单的形体，适合于进行多种材质的简单验证。

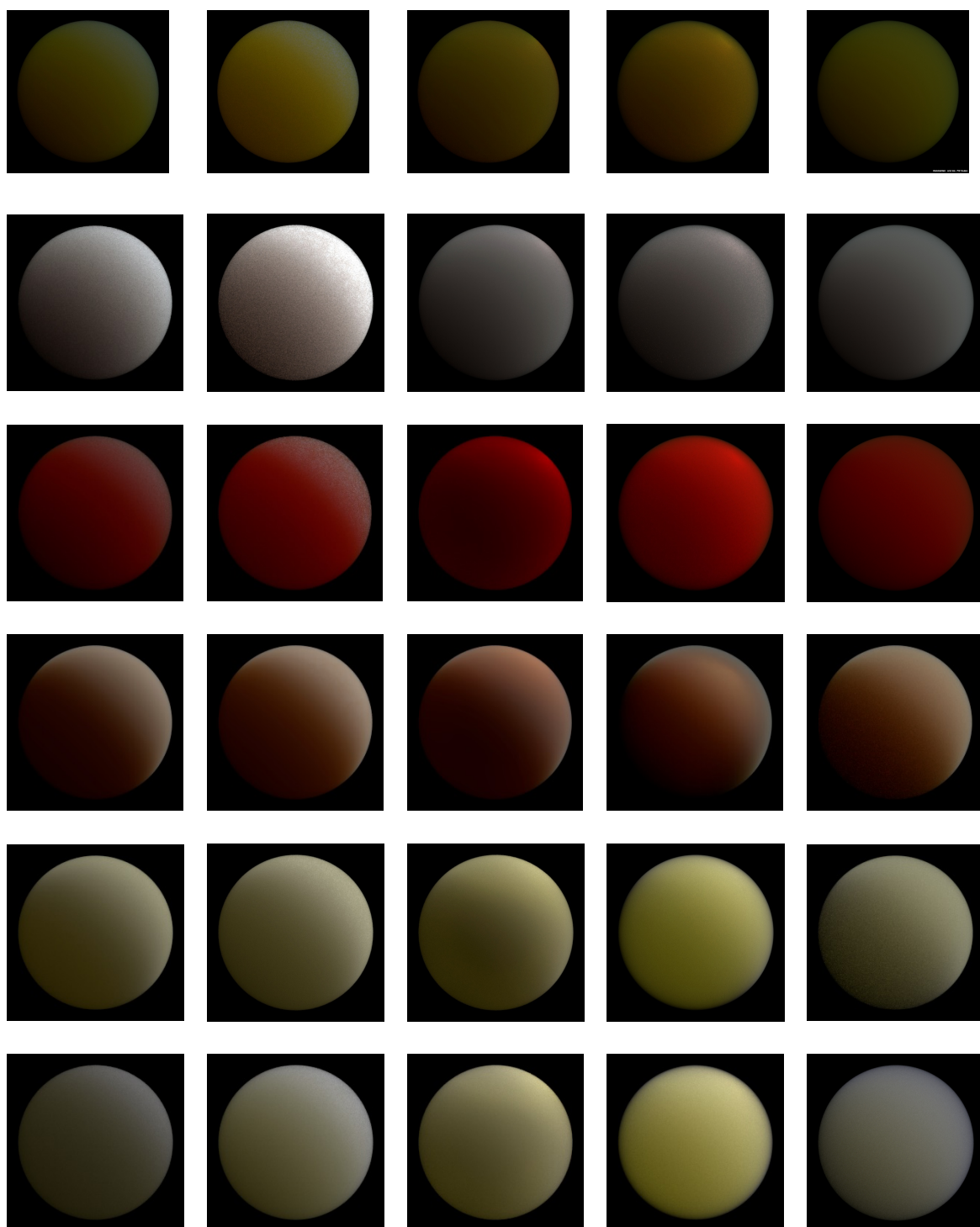


图3-8 第一列是对偶极子模型的结果，第二列是方向对偶极子模型的结果，第三列是 embssrdf 模型的结果，第四列是我们的结果，第五列是用于参照的光线跟踪结果。
 材质从第一行到最后一行分别是：Juice、Salt powder、Ketchup 、Skin1、Potato、White Grapefruit Juice

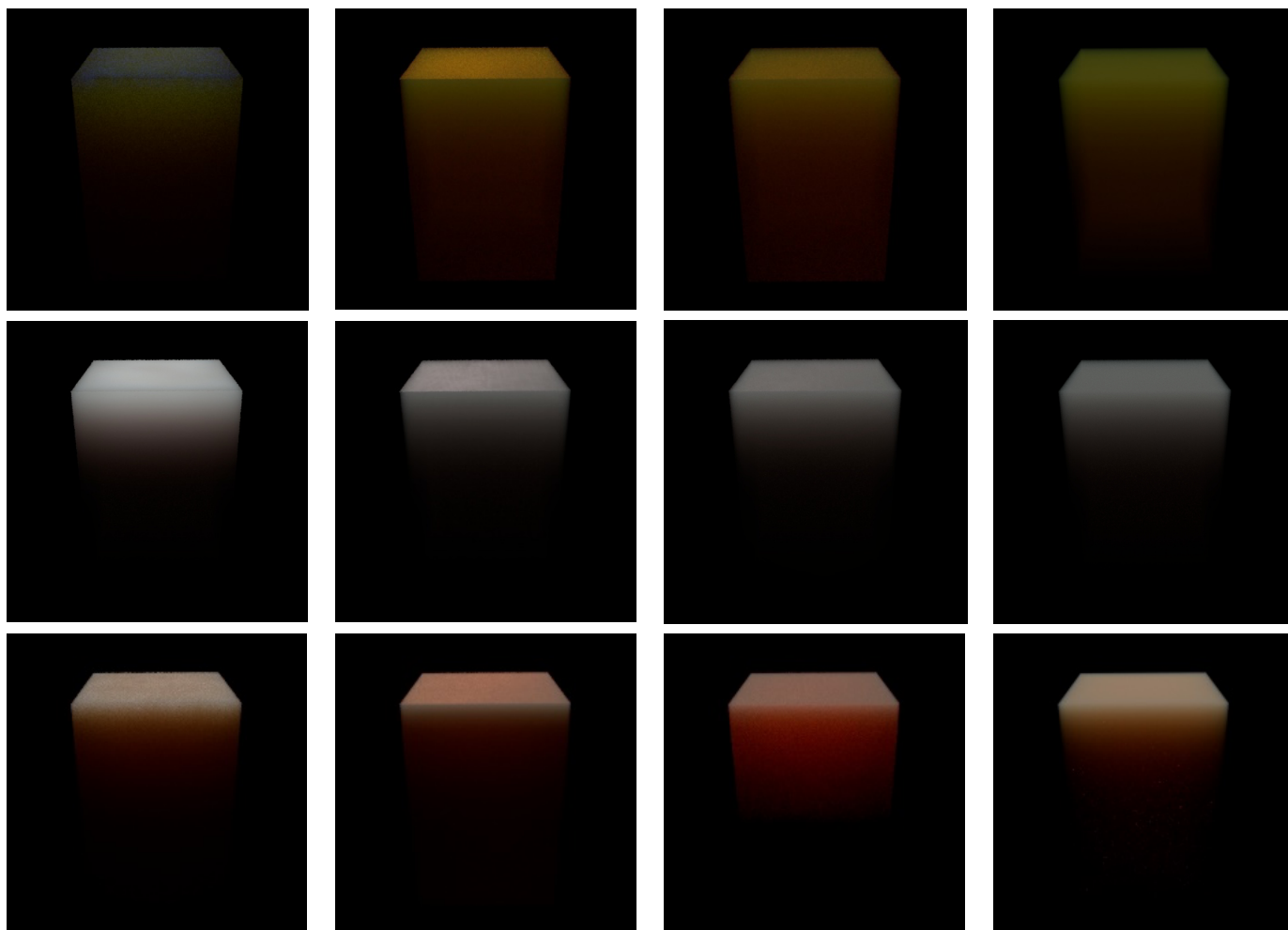


图3-9 第一列是对偶极子模型的结果，第二列是 embssrdf 模型的结果，第三列是我们的结果，第四列是用于参照的光线跟踪结果。
 材质从上到下依次是 Juice、Salt powder、Skin1。

从上面的球体、立方体渲染结果可以看出，我们的模型适用于多种材质的渲染。其中包括吸光度很大的材质：Skin1、Potato、White Grape Juice，以及吸光度较小的材质：Juice、Salt Powder、Ketchup。

我们可以观察得出，对于吸光度较大的材质，基于扩散的方法也可以取得很好的近似结果，而对于吸光度较小的材质，它们的结果则误差相对较大。

3.2.2 复杂几何体渲染结果

这里展示复杂几何体的半透明材质渲染结果，我们选择斯坦福 Bunny 模型 [20]作为渲染的模型。场景设置和之前类似，仍然采用方向光源。下面展示了具体的渲染结果：



图3-10 Bunny 模型渲染结果，从左到右分别代表 Dipole 模型、我们的模型以及光线跟踪参考。而第一行代表 Juice 材质，第二行代表 Salt Powder 材质，第三行为 Ketchup 材质。

从上面的图片中可以看出，在我们选取的几种吸光度较低的材质渲染中，我们的模型结果 RMS 要更小，也就是说我们的模型比对偶极子模型的结果要更加准确。

但需要注意的是，由于数据采集过程中假定了物体表面是无限大半平面，因此在较薄的区域、拐角处等会高估出射亮度，Bunny 的耳朵、脚趾等较薄的区域可以看出这一问题。

3.2.3 全局光照渲染结果

前面展示的结果均是局部光照下的结果，而本方法是可以直接应用于全局光照下的。本节将以经典的 Cornell box 以及 Dragon 模型为例进行展示：

cornell box 渲染结果（color bleeding）：



图3-11 Dragon 模型在环境光照下的渲染结果，材质为 Juice。

上面的结果展示了本方法用于全局光照下半透明材质的渲染效果。其中，Cornell box 中可以看出典型的全局照明效果：颜色泄漏（Color bleeding 现象）（即某个物体的颜色值不仅受到其自身着色性质的影响还会受到其周边物体的影响）；Dragon 的例子中展示了使用环境贴图作为光照下的渲染结果。

3.3 讨论和展望

我们使用神经网络对 BSSRDF 数据进行了拟合并将其实现在渲染当中，并且从拟合结果、渲染结果中证实了其有效性和准确性。

3.3.1 不足

我们的模型的缺点主要有两点：首先是渲染速度慢，由于神经网络在前向传播过程中需要计算多次矩阵的乘法运算，运行效率低，导致基于此的渲染方法速度也较慢；

表3-2 运行 100000 次 BSSRDF 求解函数的时间对比¹

对偶极子模型	Embssrdf 模型	我们的模型
0.47 ms	47 ms	15281 ms

其次是受限于原始数据的间隔，超出取值最小或者最大后预测将不再准确，比如某些物体吸光度存在大于 0.99 的情况，由于原数据中最大吸光度仅为 0.99，故这种情况下 embssrdf 模型和我们的模型得到的渲染结果的准确度会降低。

3.3.2 展望

我们的方法的主要问题在于神经网络规模较大而导致渲染速度慢以及原始数据采集的局限性（比如假定材质匀质、水平且假定是无限大半平面）。第一个问题可以考虑将矩阵乘法利用 GPU 进行加速或者缩减神经网络规模来加快其求解过程；而针对第二个问题，可以考虑在采集数据的过程中将厚度、表面曲率参数等也作为输入参数进行采样以得到更准确的数据。

¹ 测试环境为 Intel Core i7-6700K，操作系统为 Windows 10。

4 参考文献

- [1] Pharr M, Jakob W, Humphreys G. Physically based rendering: From theory to implementation[M]. Morgan Kaufmann, 2016.
- [2] United States. National Bureau of Standards, Nicodemus F E. Geometrical considerations and nomenclature for reflectance[M]. US Department of Commerce, National Bureau of Standards, 1977.
- [3] Chandrasekhar S. Radiative transfer[J]. Oxford, Clarendon Press, 1950., 1950.
- [4] Bowers J, Wang R, Wei L Y, et al. Parallel Poisson disk sampling with spectrum analysis on surfaces[C]//ACM Transactions on Graphics (TOG). ACM, 2010, 29(6): 166.
- [5] Farrell T J, Patterson M S, Wilson B. A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties in vivo[J]. Medical physics, 1992, 19(4): 879-888.
- [6] W. G. Egan and T. W. Hilgeman. Optical Properties of Inhomogeneous Materials. Academic Press, New York, 1979.
- [7] Jensen H W, Marschner S R, Levoy M, et al. A practical model for subsurface light transport[C]//Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, 2001: 511-518.
- [8] Jensen H W, Buhler J. A rapid hierarchical rendering technique for translucent materials[C] ACM Transactions on Graphics (TOG). ACM, 2002, 21(3): 576-581.
- [9] Donner C, Jensen H W. Light diffusion in multi-layered translucent materials[J]. ACM Transactions on Graphics (ToG), 2005, 24(3): 1032-1039.
- [10] d'Eon E, Irving G. A quantized-diffusion model for rendering translucent materials[C]//ACM Transactions on Graphics (TOG). ACM, 2011, 30(4): 56.
- [11] Habel R, Christensen P H, Jarosz W. Photon beam diffusion: A hybrid monte carlo method for subsurface scattering[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2013, 32(4): 27-37.
- [12] Donner C, Jensen H W. Rendering translucent materials using photon diffusion[C]//ACM SIGGRAPH 2008 classes. ACM, 2008: 4.
- [13] Frisvad J R, Hachisuka T, Kjeldsen T K. Directional dipole model for subsurface scattering[J]. ACM Transactions on Graphics (TOG), 2014, 34(1): 5.
- [14] Menon S, Su Q, Grobe R. Determination of g and μ using multiply scattered light in turbid media[J]. Physical review letters, 2005, 94(15): 153904.
- [15] Yan L Q, Zhou Y, Xu K, et al. Accurate translucent material rendering under spherical Gaussian lights[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2012, 31(7): 2267-2276.
- [16] Donner C, Lawrence J, Ramamoorthi R, et al. An empirical BSSRDF model[J]. ACM Transactions on Graphics (TOG), 2009, 28(3): 30.
- [17] Chollet F. Keras[J]. <https://github.com/fchollet/keras> 2015.
- [18] Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems[J]. arXiv preprint arXiv:1603.04467, 2016.

- [19]Jakob W. Mitsuba renderer, 2010[J]. URL: <http://www.mitsuba-renderer.org>, 2015, 3: 10.
- [20]Turk G, Levoy M. The stanford bunny[J]. 2005.

5 致谢

我的本科毕业论文是在清华大学计算机系徐昆老师和南京大学计算机系过洁老师的共同指导下完成。毕设项目的提出及解决过程中徐昆老师不仅在细节处提供诸多帮助，更重要的是其严谨的科研态度、丰富的专业知识以及灵活的研究思路给我树立了榜样。我是在过洁老师的带领下走入图形学的大门，过老师不仅耐心细致的讲解基础概念，并且引导我开始尝试科研工作，毕设项目中也得到过老师的诸多帮助。在此表达对两位老师深深的谢意。

感谢南京大学计算机系和清华大学计算机系共同为我提供的去校外完成毕业设计的机会，感谢傅忠老师、苏贵珍老师、陶先平老师以及赵超老师等的帮助。

感谢清华大学计算机系可视媒体研究中心的学长学姐的关心和帮助。尤其感谢梁缘学长、袁泰凌学长在神经网络拟合方面给予我的指点。

最后感谢评审老师的耐心阅读。

6 附录

绘制 BSSRDF 二维投影图代码:

```
import numpy as np
import Image
import colorsys
import math

def draw_data(inpt,model,res_filename):
    width,height = 500,500
    res = np.zeros((width,height))
    inputs = []
    for i in range(width):
        for j in range(height):
            tmp = list(inpt)
            xd = 2.0 * i / width - 1.0
            yd = 2.0 * j / height - 1.0
            tmp.append(xd)
            tmp.append(yd)
            inputs.append(tmp)

    outputs = model.predict(np.array(inputs))
    for i in range(width):
        for j in range(height):
            xd = 2.0 * i / width - 1.0
            yd = 2.0 * j / height - 1.0
            if xd * xd + yd * yd >= 1:
                res[j][i] = 0
            else:
                res[j][i] = outputs[i*height + j]
                res[j][i] = math.exp(res[j][i])

    res /= res.max()
    img_res = np.zeros((width,height,3))
    for i in range(width):
        for j in range(height):
            img_res[i][j] =
                np.array(colorsys.hsv_to_rgb(res[i][j],1.0,1.0))

    img_res = img_res * 255.0
    img_res = img_res.astype(np.uint8)
    im = Image.fromarray(img_res,'RGB')
    im.save(res_filename)
```