第5章

虚拟现实前沿性方向

CHAPTER 5

虚拟现实与信息技术的发展密切关联。近年来,随着人工智能等技术的发展,虚拟现实 这一概念的内涵和外延也在不断变化。目前,学术界与工业界关于虚拟现实技术的前沿性 研究主要集中于增强现实技术以及将虚拟现实与人工智能相结合等方向。本章简要介绍近 年来日益受到关注的增强现实技术,以及将人工智能应用于虚拟现实建模与渲染等研究 热点。

5.1 增强现实技术

增强现实和虚拟现实之间的联系非常紧密,是虚拟现实发展的一个扩展或者说分支,也 是近年来学术界和工业界的热点之一。增强现实系统通常综合不同领域的多种技术,包括 虚拟现实技术、计算机视觉技术、人工智能技术、可佩戴移动计算机技术、人机交互技术、生 物工程学技术等,具有鲜明的交叉特色。

尽管增强现实技术在不同应用领域内所强调的重点有所不同,而且不同档次与用途的 增强现实系统所需要的配置也各不相同,但它们在原理上存在很多共性。至于硬件设备的 具体选配要求,则往往需要用户针对经费投入、性能需求和市场效益等因素进行折中考虑。

如图 5.1 所示,一个典型的增强现实系统由虚拟场景发生器、透视式头戴式显示器、头部方位跟踪设备、虚拟场景与真实场景对准的方位跟踪定位设备和交互设备构成。其中虚拟场景发生器负责虚拟场景的建模、管理、渲染,以及对其他外设的管理。头部方位跟踪设备用于跟踪用户视线变化,实现用户观察坐标系与虚拟场景坐标系的匹配。交互设备则用于实现感官信号以及环境控制操作信号的输入输出。这里涉及的方位跟踪定位设备负责测量用户在真实环境中的六自由度位置和方位信息。从概念上来说,它与头部方位跟踪器没有区别,但实际的测量精度要求有很大差别,将直接影响增强现实系统的注册精度。

5.1.1 增强现实技术的基本内容

除了与虚拟现实技术面临共同的一些难点,比如真实感虚拟场景渲染、传感设备和交互 设备等,增强现实技术在发展中所面临的具有挑战性的难点还在于:能够将虚拟对象和真 实环境精确对准的定位手段,以及能够将虚拟场景与真实环境融为一体的显示设备。

这里对定位手段的要求不仅仅是虚拟现实系统中所要求的方位跟踪系统的精度、数据



图 5.1 典型的增强现实系统结构图

刷新频率以及延迟,更重要的是理解虚拟环境坐标系、真实环境坐标系以及用户本身视觉坐标系之间的关系,实现它们之间的精确对准以及在运动过程中维持这种对准关系,这势必要求虚拟对象在虚拟环境坐标系中的运动精确性和其相对真实环境坐标系的运动精确性。

另外,对显示设备的要求,也不再局限于用户对于虚拟环境的沉浸感,也不是虚拟环境 与真实环境的简单叠加,关键是虚拟环境深度细节、照明条件、分辨率应与真实环境诸因素 相匹配,因为不相匹配的环境会导致感觉的倾向性甚至误导性。

目前阻碍增强现实系统得以广泛应用的技术难点在于以下两方面,一方面是增强现实的显示技术,另一方面则是增强现实的跟踪注册技术。

1. 增强现实显示技术

目前,有许多的增强现实系统采用透视式头戴式显示器来实现虚拟环境与真实环境的 融合。透视式头戴式显示器由三个基本结构构成:虚拟环境显示通道、真实环境显示通道、 图像融合及显示通道。而其中虚拟环境的显示原理与沉浸式头戴式显示器基本相同,图像 融合与显示通道是与用户的最终接口,基本取决于真实环境的表现方式。因此根据真实环 境的表现形式划分,目前主要有视频透视式头戴式显示器和光学透视式头戴式显示器。

视频透视式头戴式显示器的原理如图 5.2 所示,由安装在显示器上的微型 CCD 采集外 部真实环境的图像,然后计算机通过计算处理,将所要添加的信息或图像信号叠加在摄像机 的视频信号上,实现计算机生成的虚拟场景与真实场景融合,最后通过类似于沉浸式头戴式 显示器的显示系统呈现给用户。



图 5.2 视频透视式头戴式显示器的原理

光学透视式头戴式显示器的原理如图 5.3 所示,这种头戴式显示器采用一对半反半透 镜,一方面允许外部真实环境的光透过,使佩戴者能够看到真实世界;另一方面还能反射来 自内部微型显示器的虚拟视频,将其叠加到用户的视野中,形成虚实融合的视觉效果。



图 5.3 光学透视式头戴式显示器的原理

视频透视式头戴式显示器的优点包括:合成影像较为灵活、视野较宽、实时响应速度高、虚实光照较为一致等。光学透视式头戴式显示器的优点则主要包括:结构简单、价格便 宜、分辨率高、安全性好、不需要视觉偏差补偿等。二者各有利弊,可以根据增强现实系统的 应用领域决定使用哪种透视式头戴式显示器。

2. 跟踪注册技术

在增强现实系统中,跟踪注册是关键技术之一,也是决定系统性能的重点,一直以来是 研究中的重要内容。跟踪注册精度是衡量增强现实系统性能及其实用性的关键指标。在增 强现实系统中,注册误差可分成两大类:静态注册误差和动态注册误差。当用户的视点与 真实环境中的物体均保持静止时,系统产生的误差称为静态注册误差;只有当用户的视点 或环境中的物体发生相对运动时才会出现的误差称为动态注册误差。动态注册误差是增强 现实系统中主要的注册误差,也是限制增强现实系统实现广泛应用的重要因素。

增强现实系统的静态误差主要源于以下几方面:光学系统畸变、图像噪声、机械未校 准、不正确的系统参数、算法本身以及摄像机标定时引入的误差等。系统延时是造成动态注 册误差的主要原因,因为整个系统的运行需要对用户头部位置和方向的跟踪计算、图像数据 的传输与 A/D转换、图形引擎产生虚拟增强信息,以及扫描输出至头戴式显示器正确位置, 过程中不可避免地存在一定的延迟。

现有增强现实系统中的跟踪注册技术主要有以下三种。

1) 基于硬件跟踪器的跟踪注册技术

常用的用来跟踪注册的硬件跟踪器包括电磁跟踪器、惯性跟踪器、测距仪、超声波定位 仪、全球定位系统(GPS)等。这些硬件跟踪器用来探测和跟踪真实环境中目标的位置和方 向,各有优缺点。由于这种注册技术容易受现实环境影响,所以精度较低,不适合对精度要 求较高的应用领域,如维修导引、教育培训等。

2) 基于计算机视觉的跟踪注册技术

目前,在增强现实领域中,对基于计算机视觉的注册技术的研究处于主导地位。它是通 过对一幅或多幅视频图像的图像处理,获得跟踪信息,判断固定在使用者头部的摄像机在真 实环境中的位置和方向。仅仅根据头部跟踪系统提供的信息进行三维注册属于开循环控 制,系统没有反馈增强信息与真实环境间的匹配程度,因而难以取得最佳匹配效果。而基于 计算机视觉的增强现实系统可利用图像处理和计算机视觉的方法协助注册,因此能够使注 册误差局限在像素级(以像素为单位,而不再以世界空间坐标系中的米或英寸等单位)。

3) 混合跟踪注册技术

由于目前单一的跟踪技术无法全面地解决增强现实系统中的方位跟踪问题,为此国外的一些著名大学和研究机构的研究人员提出采用混合跟踪(Hybrid Tracking)的方法对增强现实系统进行跟踪注册。所谓混合跟踪,是指采用不同种类的跟踪设备,取长补短共同完成增强现实系统的注册任务。目前常采用的硬件跟踪器包括机械跟踪器、电磁跟踪器、光电跟踪器、惯性跟踪器、超声波跟踪器、GPS等。

现有的跟踪注册方法各有优缺点和适用性。一般需要使用硬件跟踪器的增强现实系统,系统跟踪较复杂,并且会因此引入系统固有误差。基于计算机视觉的跟踪注册是一种综合性能比较好的注册方法,其通用性较强,仅通过图像分析的方式就可以得到头戴式显示器的方位信息,系统构造简单,减少了系统误差(不存在摄像机和硬件跟踪器之间的误差),尤 其适用于使用光学透视式头戴式显示器的增强现实系统,但是计算量较大,系统延迟较大。

5.1.2 增强现实技术与虚拟现实技术之间的关系

增强现实技术是随着虚拟现实技术的发展而产生的,二者间存在着不可分割的纽带关系,但在构建所需的基础技术上则稍有不同。如前所述,虚拟现实技术综合了建模、渲染、人机交互、传感等技术,力图使用户在感官上沉浸在一个虚拟环境中。而增强现实技术则主要借助显示、人机交互、传感和计算机视觉等技术,意图将计算机生成的虚拟环境或对象与用户周围的现实环境融为一体,使用户从感官效果上分辨不出虚拟部分和真实部分。基于以上的说明,二者的区别就很明显,与虚拟现实最终希望真实地模拟现实世界不同,增强现实的最终目的是利用附加信息去增强用户对真实世界的观察和感知。增强的信息既可能是虚拟的三维模型,也可能是真实物体的非几何类信息,例如路标、文字提示等。具体而言,增强现实技术与虚拟现实技术在如下几方面有较大差异。

(1)对沉浸感的要求不同。虚拟现实系统强调用户应能在虚拟环境中达到视觉、听觉、 触觉等感官知觉完全沉浸的程度,这往往需要借助于能够将用户的感官知觉与现实环境隔 离的设备,如沉浸式头戴式显示器。与之相反的是,增强现实系统不仅无须隔离现实环境, 还强调了用户在现实环境的存在感,并努力维持用户感官效果的稳定,这就需要借助能够融 合虚拟环境与真实环境的设备,如透视式头戴式显示器。

(2) 注册(Registration)。沉浸式虚拟现实系统中的"注册"强调的是呈现给用户的虚 拟环境应与用户各种感官知觉相匹配,例如当用户推开一扇虚拟的门,他所看到的就应该是 室内场景;一列虚拟火车向用户驶来,他听到的鸣笛声就应该由远及近。这里所谓的"注 册"主要是指用户的感官系统与用户本体感受之间的一致性。然而,增强现实系统中的"注 册"概念来源于计算机视觉,主要是指用户在真实环境的运动过程中维持正确的视点"对准" 关系。例如,在真实环境中叠加显示了一个虚拟的餐桌,用户在行走过程中看到的餐桌应该 随视点正确变化,并且产生与真实环境一致的光照变化。

(3) 对系统计算能力的要求不同。一般来说,即使是对现实世界周遭的简单环境进行 精确再现,虚拟现实系统也需要付出巨大的代价。这主要体现在两方面:一方面,虚拟环境 的建模与渲染本身需要巨大的人力成本与计算开销;另一方面,显示环节的实时性对数据 传输提出了苛刻的要求。当前技术条件下,虚拟环境的逼真程度还未能匹配人类的感官知 觉。相比之下,增强现实技术则要求较低,主要是因为它充分利用了周围业已存在的真实环 境,在此基础之上再扩充一些信息,这就大大降低了对计算能力的要求。

增强现实技术的应用 5 1 3

虚拟现实系统强调的是用户的视觉、听觉、触觉等感官知觉应该能够完全沉浸在虚拟环 境中。对人的感官系统来说, 虚拟现实系统是真实存在的, 而系统中所构建的物体又是现实 中不存在的,这就使虚拟现实技术更适用于许多高成本且危险的环境。因此,目前虚拟现实 主要应用于教育、数据和模型的可视化、工程设计、城市规划、军事仿真训练等方面。增强现 实技术主要是利用附加信息去增强使用者对真实世界的感官认知,因此其应用更侧重于辅 助教学与培训、辅助医疗研究与解剖训练、辅助精密仪器制造与维修、远程机器人控制、辅助 军事侦察等领域。图 5.4 展示了虚拟现实技术与增强现实技术在医学中的应用。图 5.4(a)展 示的是虚拟手术的场景,它主要用于医学生的教学培训,学生可以进入虚拟人体的内脏,检 查病灶并进行组织切片检查:图 5.4(b)展示的是增强现实辅助教学的场景,可以看到学生 在增强现实辅助信息的帮助下,使用手术器械对病人进行手术。



(a) 虚拟手术场景



(b) 增强现实辅助教学场景 图 5.4 虚拟现实技术与增强现实技术在医学中的应用

智能化建模与渲染方法 5.2

虚拟现实对虚拟对象模型的逼真度以及渲染效果的真实感要求很高。近年来,随着三 维数据采集设备的种类逐渐丰富,精度不断提升,又恰逢人工智能技术的迅猛发展,原有的 三维建模与渲染技术的智能化进程加快,虚拟对象的智能化构建日益受到重视。

5.2.1 智能化建模

智能化建模在计算机辅助设计(Computer Aided Design, CAD)/计算机辅助工程 (Computer Aided Engineering, CAE)以及其他相关工业领域一直是研究热点。通过参数 化或者非参数化的方法构造虚拟对象的外观模型,同时定义其表面材质等属性,为后续渲染 提供数据基础。传统 CAD/CAE 的方法需要大量的人力,经济成本和时间开销较大。后来 出现的基于多视图图像的几何方式重建法虽然能够较好地实现三维模型的重建,但仍然依 赖手工设计,目计算效率较低。此外,在通过主动式设备采集或重建的几何模型中,往往存 在空洞、噪声等失真现象,影响用户的感官体验。近年来,随着深度学习从二维图像域拓展 到三维空间域,基于深度学习的建模方法正成为工业界和学术界密切关注的课题之一。

根据第3章中对虚拟现实建模的分类,下面将简单介绍一下人工智能近年来在虚拟现 实形状建模与基于物理的建模等方向上的发展状况。

1. 智能化的形状建模

近年来,随着深度学习在几何领域的兴起,基于深度学习的形状建模正成为研究热点, 并逐渐形成了"几何深度学习"这一理论分支。目前,形状建模的研究主要集中于从图像重 建虚拟对象的几何模型,以及对几何模型进行优化等方面。其中,根据几何模型的表示方 式,又可分为显式表示法与隐式表示法两类。

1) 显式表示法

显式表示一般是指将虚拟对象表示为点云或多边形网格等形式。多视点图像是三维 几何模型的一种有效表示方式。利用多视点图像生成多视点深度图像,进而可以生成虚 拟对象的外表面,可以有效地重建虚拟对象的几何模型。因此,可以将虚拟对象的生成 问题转换为从多视点图像生成几何模型的问题。目前,典型的从多视点图像重建几何模 型的网络一般由编码器和解码器两部分组成。其中,编码器接收物体类别 c、视点 v 和变 形参数 θ。这三类输入数据通过若干全连接层后,得到隐空间的特征向量。解码器则将 特征向量通过若干反卷积层,得到模型的 RGB 图像和对应的目标分割图。通过最小化网 络生成的 RGB 图像以及分割图与真实值的误差,即可训练出一个具有良好重建效果的学 习模型。

2) 隐式表示法

基于图像的视点合成是构建虚拟场景的经典技术之一。随着神经网络的发展,神经网络作为几何模型的隐式函数,可以达到不同的应用目的,如新视点生成、虚拟视点生成以及重光照等。常用的几何模型神经网络隐式函数表示方法包括占位网络(Occupancy)、符号距离函数(Signed Distance Function,SDF)等。例如,像素对齐隐式函数(PiFU)通过神经网络,根据输入的单视点图像或多视点图像,学习到人体外表面的隐式函数。在训练阶段,输入单张图像,通过两个自编码器分别学习人体表面的概率分布并推断可能的纹理,同时实现视点变换矩阵的学习。在测试阶段,输入一张图像,首先通过神经网络得到占位网格表示,然后利用 Marching Cube 算法重建外表面,最后再利用 UV(紫外光线)技术实现纹理重建。

2. 智能化的物理建模

目前对于虚拟对象的物理建模,按照所研究的对象类型,可大致分为人体运动建模、柔 性体建模与流体建模三类。

1) 人体运动建模

对人体运动建模来说,主要是使用神经网络来进行运动数据帧的合成。通过对原始运动数据对齐并设置标签,把预处理后的数据向量化,并将向量化结果作为神经网络的输入进行编码,学习网络权重。通过编码-循环-解码(Encoder-Recurrent-Decoder,ERD)模型,在循环层的前后加入非线性的编码和解码网络,用于识别和预测人体运动的姿态。将特征表示与运动时序动态结合,ERD模型在隐变量空间使用长短期记忆(LSTM)网络模型来预测人体运动的下一个姿态。

2) 柔性体建模

对柔性体建模的主要思想是利用深度学习进行虚拟对象的变形。例如,使用基于网格 变分自编码器进行衣物等柔性体变形。该方法利用深度神经网络探索虚拟对象的潜在形状 空间,并且能够通过对现有虚拟对象的变形生成原始数据集中不存在的新的虚拟对象。为 了有效表示任意的网格模型,该框架使用变分自编码器和旋转不变网络的曲面表示方法,并 使用全连接的网络架构和简单的基于均方差的重构损失函数。

3) 流体建模

在传统的基于欧拉网格法的流体建模框架中,投影步求解过程往往是计算资源和计算 时间消耗占比最大的一部分,特别是对于高精度、高分辨率的大规模流体场景,传统的流体 建模算法不论是在计算速度,还是在模拟效果上,均存在明显不足。随着人工智能技术的深 入发展,深度神经网络模型以其强大的数据学习能力,被广泛应用至计算机图像分类、语音 识别、流体细节合成等研究领域,其稳定、高效的计算模式,为计算机流体模拟提供了一种新 的问题解决途径。

5.2.2 智能化渲染

虚拟场景的真实感渲染是指通过计算机模拟光线在 3D 场景中传播的物理过程,将设 计者创作的由视点、光源、三维几何形状、动画、材质等组成的场景转换为高度真实感的连续 帧画面。全局光照能够模拟真实光线在场景中传播的过程,对渲染真实感具有至关重要的 作用。而全局光照技术的核心在于渲染方程,该方程能够计算光束和三维物体的交互过程 中的能量传递,具有全局性、递归性等特点,可以使整个虚拟场景在光照传播过程中保持能 量守恒。全局光照计算需要耗费大量的时间和计算资源,虽然目前可以加速渲染的方法有 很多,但其依然是高度真实感渲染领域的瓶颈。

近年来,人工智能技术的发展激发了大量基于神经网络的真实感渲染的相关研究。这些研究分别集中于解决真实感渲染流程中的某个问题,并取得了很好的效果。例如,一些研究者提出了轻量级的材质建模方法,可以替代工业界传统采用的复杂的材质制作流程;一些研究者还利用数据驱动的机器学习方法大幅提升了全局光照的计算速度。此外,许多商业渲染器已经集成了基于机器学习的后处理降噪技术。但是,目前机器学习方法并没有在所有真实感渲染领域得到应用,对于一些特殊效果的渲染,例如毛发、布料、复杂光源的渲染,依然有待于寻找机器学习解决方案。

下面简要介绍人工智能技术在全局光照计算优化、参与介质渲染优化等方向的进展。

1. 智能化全局光照计算

全局光照算法同时考虑了直接来自光源的光线(直接光照)和经过其他表面反射的光线 (间接光照)。使用全局光照算法可以模拟现实世界中的大部分光照效果,如阴影、环境遮 挡、反射、焦散、次表面散射等现象。然而,全局光照计算十分耗时,通过使用机器学习方法 加速计算是目前的研究热点。

1) 基于预计算辐照度的间接光照计算

加速计算全局光照的一种常见做法就是预先计算出虚拟场景的全局光照信息(如辐射度),并将其存储于三维空间的几何形状上。这些预存储的信息会在视点改变时再次被利用,从而避免重复昂贵的光照计算。例如,构建一种辐射度回归函数(Radiance Regression

Function, RRF), 随后使用机器学习方法将其建模为一个多层神经网络, 用以快速、实时地 计算全局光照效果。该函数可以根据表面上每一点的视点方向与光照条件, 计算出该点的 间接光照信息。

在使用 RRF 进行实时渲染时,首先进行直接光照的计算,同时获取每个表面点的属性, 然后由基于神经网络的 RRF 模型计算得到每个表面点的间接光照信息,并与计算的直接光 照信息进行合成,得到最终的全局光照结果。该方法可以实时渲染出带有全局光照效果的 分辨率为 512 像素×512 像素的结果图片,并保持渲染 30fps 以上的速度,可有效渲染出包 括焦散、高频的反射、间接硬阴影等复杂的光照效果。但是,该方法只适用于静态几何场景, 允许光源和视点变化。

2) 基于机器学习的路径指导方法

使用机器学习或统计学习方法优化光路传输也是实现快速计算全局光照效果的途径之一。传统路径跟踪或者双向路径跟踪方法,在进行 BSDF(双向散射图分布函数)或者光源 采样时,通常只考虑到局部的信息,而没有考虑到全局的信息,因此导致在复杂光路时,得到 一些对整体贡献较少的路径,从而使得渲染噪声较大。路径指导则是通过一定的方式来获 取光路中的更多全局信息,并且根据这些信息来指导重要性采样,从而达到减少噪声的目 的。近几年,研究者开始关注基于机器学习的路径指导方法,该方法对于全局光照加速计算 效果显著。

3) 光照采样算法的优化

利用大规模数据集,针对首次反弹入射辐射场的自适应采样和重建训练神经网络,可以 有效优化现有光照采样算法的时间性能。该方法结合了基于深度强化学习(DRL)的质量 网络(Quality Network,Q网络)和基于 4D 卷积神经网络的重建网络(Reconstruction Network,R网络)。Q网络可预测和指导自适应采样过程,R网络可重构 4D 空间中的入射 辐射场。

2. 智能化参与介质渲染

在真实场景中,随时可见各式各样的参与介质(如蜡烛、牛奶、橄榄油、烟雾等),光线在 参与介质中传递时会被吸收或者发生散射。为了模拟介质中的散射现象,需要花费大量计 算时间。特别是由异质密度形成的可见结构(如大气云)更加挑战渲染算法的效率。参与介 质的表观通常是由成千上万的内部光子相互作用而成,即使是忽略它的离散性质,用连续体 积近似模拟,通过求解辐射传递方程来估算光传输仍需大量计算。而基于预计算的方法将 特定的材质的散射信息用表存储起来,渲染时直接从预存表中提取对应的辐射度值。但是 这类方法耗费大量的表存储空间。故目前参与介质渲染方法在互动性、即时性和吞吐量大 的应用需求中尚存在较大瓶颈。一些方法通过简化计算来提升渲染速度,例如近似值扩散 理论、密度估计、半解析解等方法,但是这些方法大大降低了渲染质量。

近年来,有学者提出使用蒙特卡洛积分和神经网络相结合来高效地渲染大气云的技术 RPNN(Radiance-Predicting Neural Networks)。该方法从大量样本中提取采样点对应的相 关着色位置和光源的几何信息,再将这些信息输入深度神经网络中,以得到最后的辐射度。 研究者们使用了一个基于点模板(stencils)的层次结构。其中,每一层模板的覆盖范围是上 一层的 8 倍。这样输入信息不仅能够表示微小的细节,同时又能有大气云的整体形状信息。 该方法使用一个 10 层的渐进式学习神经网络来学习逐点的辐射度,每一层的输入来自上一 层神经网络和新的点模板。该方法在保证渲染质量的同时,渲染速度提高了千倍。

5.3 智能化建模实例——基于深度学习的流体建模

5.3.1 背景知识

如前所述,使用传统方法进行高分辨率流体建模需要大量的计算资源以及时间。为加 速仿真过程,基于深度学习的流体建模方法进入了人们的视野。和传统流体建模方法类似, 基于深度学习的流体建模方法也分为两大分支,即由拉格朗日法衍生出的光滑粒子流体动 力学(Smoothed-Particle Hydrodynamics,SPH)方法,以及由欧拉法衍生出的离散网格法。 前者的核心思想是将 SPH 中粒子各种属性的求解问题转换为粒子速度以及位置的回归问 题,可使用随机森林等模型进行训练;后者则主要针对规则网格中的每一个网格的属性进 行独立处理,利用全连接神经网络编码当前位置的特征,包括压力、速度的梯度以及边界条 件等。与 SPH 方法相比,离散网格法可以产生连续的输出,进行更加细致的建模。

下面将介绍一种基于卷积神经网络(Convolutional Neural Networks, CNN)的离散网格流体建模方法。

5.3.2 基于深度学习的流体建模方法

流体建模主要围绕三大变量进行:密度场、速度场、压力场。其中,密度场的变化可以 通过使用速度场进行平流导出,对速度场进行压力投影操作可以计算压力场,通过压力场并 根据牛顿第二定律可以对速度场进行更新。因此,速度场与压力场在流体仿真过程中驱动 流体的运动,而密度场则更加容易观测。

使用基于深度学习的流体建模方法时,直接拟合密度场较为困难,这是因为流体作为不 定形物体,不存在较强的形状先验知识。因此,构造训练数据集时,需要遍历各种密度场的 分布情况以及各种训练参数才可以对虚拟场景有较好的覆盖,而为了拟合海量的数据,需要 大量网络参数以及漫长的模型训练时间。网络参数量越大,在进行推理时,就会拖慢推理速 率,从而降低使用深度学习方法相较于传统方法的优势。使用深度学习的方法拟合速度场 也较为困难,因为速度场在仿真的过程中受外力、黏性等影响,不够稳定。因此,这里使用散 度场作为输入,利用 CNN 拟合压力场的方法进行流体建模。整个算法大致分为流体建模 数据集的构建以及 CNN 流体求解模型的构建与训练两部分。

1. 算法实现

1) 流体建模数据集的构建

为了尽可能地覆盖真实世界的流体种类,创造多样化的流体训练数据,使用流体仿真系统 Mantaflow 进行离线流体数据的生成,并采用以下策略生成丰富且贴合实际需求的流体数据集:

(1) 使用随机的小波湍流噪声(Wavelet Turbulent Noise)初始化空间中的速度场。

(2) 在空间中随机放置一些基本几何体与真实世界中常见的物体作为空间障碍物。

(3) 仿真时,在空间中随机添加局部噪声。

(4) 在仿真空间中的随机位置, 添加随机大小、随机强度并且发射时间随机的速度源, 以便增加仿真空间的扰动。 在初始化速度场时,使用小波湍流噪声进行初始化。为了保证初始速度场尽可能地无 散,采用首先初始化低分辨率的小波湍流噪声,然后进行插值到高分辨率,再进行多帧仿真 的方式生成初始无散速度场。因为小波湍流噪声会有细微的散度,若使用其进行初始化,会 影响后续训练,通过进行从低分辨率到高分辨率的插值,可以有效降低空间中的散度,插值 后进行 10 帧的仿真,可以进一步稳定并消除散度。为了保证数据的多样性,初始小波湍流 噪声的控制变量(噪声强度、噪声的相对缩放、噪声位置等)都是随机产生的。

空间障碍物分为两类,一类为基本几何体的模型,一类为真实生活中常见对象的 3D 模型。这里使用的基本几何体包括球体、正方体等;真实世界对象的 3D 模型则取自 NTU 3D 数据集,其典型例子如图 5.5 所示。



图 5.5 NTU 3D 数据集中的模型

在进行流体仿真数据生成时,使用 0.125s 作 为时间迭代步长,总共进行 256 帧的仿真,每 8 帧 记录一次当前的流体状态。记录的信息包括:流 体当前状态的速度场、压力场以及空间障碍物的信 息。对于 2D 仿真数据生成,设置仿真空间分辨率 为 256 像素×256 像素,一共生成 320 个流体仿真 场景,每个场景 64 个训练数据,总大小为 26GB。 对于 3D 仿真数据生成,设置仿真空间分辨率为 64 像素×64 像素×64 像素,一共生成 320 个流体 仿真场景,每个场景 64 个训练数据,总大小为 450GB。图 5.6 展示了在 2D 空间得到的流体仿真 数据,颜色的深浅表示该点速度的大小,越偏向深 色,速度越小,越偏向浅色,速度越大。白色部分表 示位置被障碍物占据,不存在流体。



图 5.6 2D 空间流体仿真数据展示

2) 基于 CNN 的流体建模

CNN 流体建模网络可以表示为下列公式:

Þ

$$= f_{\rm conv} (\nabla \bullet u_{\rm div}, o_t)$$

(5.1)

其中, f_{conv} 表示深度学习模型, 其输入为当前速度场 u_{div} 的散度场 ∇ • u_{div}, 以及当前网络 仿真空间的障碍物边界情况 o_t, 输出为仿真空间中的压力场 p。使用压力场对有散速度进 行速度更新, 可获得接近无散的速度场, 通过极小化更新后的速度场散度可对网络进行训 练。为了节省参数,并且保证求解结果在全局以及局部上都较为恰当,选择基于 CNN 的多 尺度网络对输入数据进行处理。流体仿真网络结构图如图 5.7 所示。



图 5.7 流体仿真网络结构图

该网络的计算分为4步。

(1) 计算输入的散度场**∇**•*u*_{div} 的标准差,并根据标准差将输入的散度场进行归一化。 使用深度学习进行流体建模的一大难点是流体速度场并没有统一的流速尺度,不同的流体 流动场景中,流体的流速可能会有上百倍的差距,而一般的深度模型只能对一定范围的数据 进行拟合,对于超出响应范围的数据输入往往不能产生理想的效果。因此,通过归一化,对 不同尺度的输入归一化到同一分布,让输入更加利于网络学习,从而降低网络结构的复 杂度。

(2)将输入的归一化散度场以及虚拟场景中障碍物的标记矩阵分别进行 2 倍与 4 倍的 下采样,并使用共享参数的卷积子网络 conv subnet1 对其进行处理。该方法使得网络在不 同尺度上都有较为合理的输出,同时也有效降低了网络参数以及计算代价。为了获得全局 较为合理的结果,需要极大地增加其卷积有效感受野的范围,这无形中就增加了网络的参数 量以及训练与预测时间,而使用共享参数的多尺度网络对其进行预测处理,则可以在保留细 节的同时,获得更大网络的感受野范围,从而获得较为理想的结果。子卷积网络 conv subnet1 是具有 3 层结构的全卷积网络,每层包含卷积操作、批量归一化操作以及激活 操作。

(3)为对多尺度网络的输出分别进行1倍、2倍、4倍的上采样到输入的大小,然后通过 第二个子卷积网络 conv subnet2得到归一化的输出结果。该步骤通过对多尺度输出进行 综合,得到了兼顾全局以及局部归一化的压力场输出结果。conv subnet2是一个4层卷积 神经网络,具有与 conv subnet1相近的结构,每一层同样包含卷积、批量归一化、激活等 操作。

(4) 对归一化的压力场输出结果使用网络输入时的归一化参数进行还原,获得与输入 量级相匹配的压力场预测结果。

网络通过极小化速度更新后的散度进行训练,损失函数可以表示为下列公式:

$$\log = \frac{1}{n} \sum w_i | \nabla \cdot u_i |^2$$
$$= \frac{1}{n} \sum w_i \left| \nabla \cdot \left(u_{\text{div}} - \frac{1}{\rho} p \right) \right|^2$$
(5.2)

其中, $\nabla \cdot \left(u_{\text{div}} - \frac{1}{\rho} p \right)$ 为速度更新后的散度, w_i 为对位置*i*的误差的加权。该训练方法有两点优势:首先,该方法可以实现了完全的无监督学习,并不需要训练数据的标签,对数据集的生成要求较低;其次,该方法可以使用网络自己生成的数据进行训练,从而可以让网络自动纠正学习过程中的产生的微小偏差。

因为带有边界条件的流体训练数据相较于无边界条件的流体训练数据较为稀少并且受限于流体建模网络的容量,网络会在边界条件下拟合效果不佳,对此使用误差加权参数 w_i 对不同位置的误差进行加权,加强网络对存在边界条件的样本的学习。w_i 的定义如下列 公式所示:

$$w_i = \max(1, k - d_i) \tag{5.3}$$

其中,d_i为位置i的距离场值,定义障碍物的内部的距离场为0,障碍物外部的距离场值是 该位置到障碍物的最近的距离。该公式的含义是对于靠近障碍物的样本,线性增大其训练 的权重,提升网络在边界条件下的表现效果。

仅使用单帧的流体仿真数据进行独立的训练,每一帧的输出可能存在细小的偏差,经过 多帧仿真,会产生误差积累的问题:训练时使用的训练数据是由传统流体仿真算法产生的 无散高质量数据,而随着网络建模的进行,误差会慢慢积累,从而导致速度场散度逐渐增大, 训练数据分布与仿真时输入数据的分布差距逐渐加大,网络输出不稳定,最终无法获得较长 时间序列上的稳定建模结果。为了让基于深度学习技术的流体建模可以在较长的时间内取 得相对稳定的结果,应采用网络自动纠偏方法提升网络解算结果的稳定性。网络在训练时, 不仅通过极小化当前帧的散度加以训练,还对使用当前网络进行 n 帧的后向仿真,并使用 极小化 n 帧仿真后的速度场散度的方法实现网络的自动纠偏。

2. 计算结果

1) 参数设置

训练过程中的超参数包括每次训练使用的 Batch Size、迭代轮数、网络学习率、使用网络自动纠偏算法的概率以及其向后建模迭代的步数、每次网络求解的时间步长、建模空间的大小等。

对于 2D 流体建模网络训练,Batch Size 为 64,并对数据集进行 200 轮训练,网络学习率 设置为 5e-5,训练时运行网络自纠偏算法的概率为 0.9。每次运行自纠偏算法时,以相等 概率进行 4 步或 16 步后向仿真,并使用其结果作为训练数据,对于边界条件位置的损失函数 加权值 k 设置为 2。流体的迭代步长为 0.1s,并在 128 像素×128 像素的空间中进行训练。

对于 3D 流体建模网络的训练,因为 3D 包含着更多的信息,使用 128 像素×128 像素× 128 像素的仿真空间会超过当前的硬件限制。因此,将 3D 仿真训练的分辨率降为 64 像素× 64 像素×64 像素,并且 Batch Size 设置为 32。网络学习率设置为 5e-3,其余参数保持不变。

2) 2D 流体建模结果展示与分析

在进行 2D 建模时,实验设置空间分辨率为 128 像素×128 像素,并且设置初始密度源

位于空间底部,直径为 38 像素,其发射密度的初始速度场为 1 像素/s,并且初始喷射密度也 恒定为 1g/像素,迭代步长为 0.1s。

图 5.8 和图 5.9 分别展示了使用传统的预处理共轭梯度法(Preconditioned Conjugate Gradient, PCG)和本节算法,在相同仿真条件下对流体的建模结果。两幅图第一行的三幅 子图分别展示了密度场、x 方向速度场、y 方向速度场的图像。第二行的两幅子图分别是压力场图与散度场图。



图 5.9 基于 CNN 的 2D 流体建模结果展示

如图 5.9 所示,在无障碍物、无噪声的情况下,建模结果呈现标准的对称状,并且从 x 方向速度场可以看出由于流体由下往上流动,上部中间的密度场有向两侧扩散的速度,并且 为保证流体的不可压缩性质,下面两侧的速度场有向中间运动的趋势。因为流体在推动物 质向上运动,因此上方的压力较大,与之相对应,下方的压力较小。散度场图显示整个建模 空间的散度都控制在较小的范围内。与使用传统 PCG 流体求解器的结果相比,基于 CNN 的方法的建模结果与其总体相近,但在流体边界细节上仍有一定缺失,图中用白色框显示。 图 5.10 进一步展示了在遇到正方形障碍物之后的流体建模结果。



图 5.10 2D 流体与正方形障碍物的交互展示

如图 5.10 所示,在流体接近障碍物的区域都呈现较大的压力,压力迫使速度场向障碍 物两侧进行扩散,并且流体密度场并未与障碍物发生明显碰撞,绕开了障碍物继续向上扩 散。由此可见,利用本节算法,在面对障碍物时仍能够产生较为合理的结果。

3) 3D 流体建模结果展示与分析

进行 3D 建模时,设置空间分辨率为 64 像素×64 像素×64 像素,并且设置初始密度源 位于空间底部,形状为半径为 10 像素的圆形,发射密度的初始速度场为 1 像素/s,并且初始 喷射密度也恒定为 1g/像素,迭代步长为 0.1s。

图 5.11 展示了 3D 流体建模的结果,同样也与传统 PCG 流体求解器进行对比。



图 5.11 3D 流体建模结果展示



图 5.11 (续)

如图 5.11 所示,从上到下分别为基本烟雾建模、与球体障碍物交互结果,以及与正方体 障碍物交互结果;从左到右则分别为 PCG 流体求解器求解结果的渲染图像、PCG 流体求解 器求解结果密度场的中心垂直切面可视化、本节算法的渲染图像,以及本节算法求解结果的 密度场中心垂直切面可视化。在没有障碍物与噪声干扰的情况下,流体的密度场呈现对称 形态,建模效果较为稳定。球体、正方体作为障碍物的建模结果显示本节算法在复杂条件下 也可以较好地应对复杂的边界情况,但是在细节生成与求解对称性等方面仍较传统流体求 解方法有所欠缺,主要体现在流体边界较为模糊(图中以白色实线框显示)以及对称性不佳 (图中以白色虚线框显示)两方面。

4) 建模效率分析

基于深度学习的流体建模,其最大优势在于时间效率。表 5.1 展示了基于 CNN 的方法与传统 PCG 流体求解器的速度对比。

	PCG 流体求解器建模总时间	深度学习方法流体建模总时间
$2D(128px \times 128px)$	0.04s(25fps)	0.006s(180fps)
$3D(64px \times 64px \times 64px)$	0.3s(3fps)	0.0125s(80fps)

表 5.1 基于深度学习的流体建模方法与传统 PCG 方法速度对比

上述结果是在 28 核心的 Intel Xeon Gold 6130T CPU 以及 Nvidia Titan V GPU 的机器上得到的。由表 5.1 可见,基于深度学习的方法在 2D 与 3D 的流体建模任务中相比传统方法都取得了巨大的进步。

5.4 智能化渲染实例——三维积云的可微渲染

5.4.1 背景知识

1. 可微渲染

近年来,可微渲染成为学术界研究的热门方向。可微渲染的本质是传统图形渲染的逆向过程。传统意义上的渲染过程是指根据三维几何信息渲染得到二维图像的前向过程。可微渲染与之相反,它是根据传感器、摄像机等物理设备获取的数据去逆向推测渲染所需的几何、光照条件、材质,以及运动等参数,以便图形渲染器可以准确地还原真实的渲染场景。因此,可微渲染实际上是从二维图像恢复到三维场景的逆向过程,其核心任务是构建渲染图像与模型参数之间的关系。

根据所要恢复的虚拟场景的数据表示方式,可将现有可微渲染方法分为基于点云、体素和网格三类方法。由于三角形网格能够更好地表示虚拟对象的拓扑关系,目前主流方法大

多采用三角形网格作为研究对象。从技术路线的角度,现有方法又可分为解析导数法、近似 渲染法与全局光照法。

解析导数法是指利用相邻三角形的渐变过程或者使用连续的渲染函数代替传统渲染过 程中离散的部分,例如光栅化过程。其主要目的是使得整个渲染过程连续可导,为使用深度 学习的方法训练求解渲染所需的参数提供便利。

近似渲染法主要针对光线追踪或光线行进算法,将被遮挡的虚拟场景部分的信息也加 入正向渲染流程中。例如,利用基于概率分布的"软光栅化"代替标准的光栅化中基于深度 缓存的可见性判断与裁剪等操作,使最终投影到屏幕上的每个像素的每个三角形面片都能 以一定的概率贡献其颜色。通过这种方式,可以避免传统渲染过程中因可见性判断而出现 不可微的问题。

全局光照法即利用蒙特卡洛估计法计算每个像素的颜色值。针对蒙特卡洛估计中光线 反复求交运算导致计算量庞大,以及光线在物体边界处的不连续项无法求导的问题,通过一 些数值优化方法使整个渲染过程可微化。例如,利用积分参数化来近似不连续部分的导 数等。

对参与介质的辐射传输方程求导,实现对体渲染的逆向求解,这样的渲染器不仅支持三 角形网格的刚性物体,而且支持参与介质的体渲染过程。然而,由于次散射过程的复杂性, 每一次渲染计算开销都非常大,计算效率低下导致优化效果较差。

2. 参与介质的辐射传输方程

空气中特定粒子的物理属性决定了它与光线的相互作用效果,不同的粒子作用效果也 不同。通常,光线在云层中发生常见的传播情况主要有4种:吸收、向外散射、向内散射、自 发光。

如图 5.12(a)所示,当光子穿过云层时,它的能量会被吸收一部分,这部分被吸收的能量沿着 ω 方向的变化量在单位距离的微分形式如下列公式所示:

$$\frac{\mathrm{d}I}{\mathrm{d}z} = -k_a(x)I(x,\omega) \tag{5.4}$$

其中, $I(x,\omega)$ 是在 x 点处沿 ω 方向的辐射强度, $k_a(x)$ 表示在 x 点的吸收系数, $\frac{dI}{dz}$ 表示辐射值在距离 z 范围内的微分形式。所以,在单位距离 dx 中对于辐射强度的吸收值为 $k_a(x)$ dx。 而且, $k_a(x) = \sigma_a \rho(x)$,其中 σ_a 是参与介质的吸收截面, $\rho(x)$ 是所处位置的密度。同理,如 图 5.12(b)所示,光子还会发生向外散射,散射系数为 $k_s(x)$,辐射强度为 $I(x,\omega)$ 的向外散 射过程的微分形式与公式(5.4)类似,向外散射系数 $k_s(x) = \sigma_s \rho(x)$,其中 σ_s 是参与介质的 散射截面。由于吸收与向外散射的物理作用都是对辐射强度的衰减,所以通常可以把两者 合并,即 $k_t(x) = k_a(x) + k_s(x)$, $k_t(x)$ 也被称为消光系数。

如图 5.12(c)所示,光子不仅会减弱,同样也可能会被加强,这就是向内散射过程,即周围的光散射进来一定辐射,向内散射项目系数为 k_s(x)。最后,对于其他特殊物质,如果粒子可以进行自发光,如烟花、岩浆等,如图 5.12(d)所示。这也是一种辐射量的增强,并以 k_a(x)的系数将周围的辐射强度吸收进来。但是针对积云渲染时,可以忽视这种情况。综上所述,积云渲染模型在单位路径 z 下的微分形式如下列公式所示:

$$\frac{\mathrm{d}I(x,\omega)}{\mathrm{d}z} = -k_a(x)I(x,\omega) - k_s(x)I(x,\omega) + k_s(x)I_s(x,\omega)$$
(5.5)



图 5.12 光线与参与介质的 4 种交互情况

公式(5.5)给出了介质中某一点辐射强度的变化情况,那么如果想得到最终射到摄像机的辐射强度,就必须将某一点的微分形式转换成路径的积分形式,如下列公式所示:

$$I(x,\omega) = \int_{0}^{z} \tau(x,y) [k_{a}(x)I_{s}(y,\omega) + k_{s}(x)I_{s}(y,\omega)] dy$$
(5.6)

$$\tau(x,y) = e^{-\int_{0}^{y} (k_{a}+k_{s}) ds}$$
(5.7)

$$I_{s}(y,\omega) = \int_{S^{2}} p(\omega,\bar{\omega}) I(y,\bar{\omega}) d\bar{\omega}$$
(5.8)

其中, $I(x,\omega)$ 表示最终的辐射强度,y表示路径中的任意一点, $L_s(y,\omega)$ 表示内散光,由该 点的射入光线的立体角积分形式给出, $p(\omega, \bar{\omega})$ 表示相位方程, S^2 表示该点的立体角面积。 $\tau(x,y)$ 表示从x点到y点这段距离光的损失比例,由消光系数在光学厚度上的积分给出。 下面详细介绍 $\tau(x,y)$,如图 5.13 所示,一束光线经过参与介质并从 x_a 处射出,经过上述 的物理作用之后射出的辐射强度遵循比尔定律,如下列公式所示:

$$I(x_{a},\omega) = I(x_{b},\omega)e^{-\int_{D_{2}}^{D_{1}}k_{t}(x_{u})du} = I(x_{b},\omega)\tau(x_{b},x_{a})$$
(5.9)

其中, $\int_{D_2}^{D_1} k_t(x_u) du$ 表示光学厚度, 它指的是消光系数在路径上的积分, 积分项提取成 $\tau(x_b, x_a)$, 表示 x_b 与 x_a 两点之间的透光率。



图 5.13 辐射传输过程示意图

此外,辐射强度由于自发光和向内散射而增强。为了导出辐射传输方程,必须考虑辐射 增量。光在参与介质中的传输过程由辐射传输方程表示,如下列公式所示:

$$I_{\text{out}}(x,\omega) = \int_{D_2}^{D_1} \tau(x_u, x_a) k_t(x_u) J(x_u, \omega) du + I_{\text{in}}(\omega) \tau(x_b, x_a)$$
(5.10)

其中, x_u 表示距离 D_1 到 D_2 之间的任意一点,通过对这段距离进行辐射强度积分(前一项) 以及计算背景光(后一项),得到了光线射出参与介质并进入人眼的辐射强度 I_{out} 。其中的 $\tau(x_u,x_a)k_t(x_u)$ 已经由上述给出, $J(x_u,\omega)$ 表示辐射强度的来源,它的具体组成如下列公式所示:

$$J(x,\omega) = J_{e}(x,\omega) + J_{ss}(x,\omega) + J_{ms}(x,\omega)$$
(5.11)

辐射来源 $J(x,\omega)$ 由三项组成, $J_{e}(x,\omega)$ 描述由于参与介质的自发光而产生的辐射强度,由于参与介质并无自发光的特性,暂时不用考虑这方面的辐射强度, $J_{ss}(x,\omega)$ 表示由于 光源的单次散射而增加的辐射强度, $J_{ms}(x,\omega)$ 表示由于多次前向散射而增加的辐射强度。

公式(5.10)虽然给出了辐射强度的积分形式,但是在实际渲染的过程中,不可能求得准确的积分结果,只能利用离散形式进行估计。传统体渲染方法会用到蒙特卡洛估计法。如上所述,蒙特卡洛估计法就是在一条路径上进行多次采样,已知辐射强度的积分形式,就可以对每个采样点进行辐射强度的计算。从摄像机发出多条光线,每条光线穿过虚拟场景,遇到虚拟对象发生折射和反射,完成递归的过程。对于积云场景,需要在一条光线路径上进行蒙特卡洛采样,并计算每个采样点的辐射强度,将每一个点的辐射强度除以该采样点的概率密度函数,得到该路径的实际辐射值,然后进行下一次递归,直到超过最大求交次数。从以上描述可见,传统体渲染方法计算复杂度高,时间开销较大。

下面将介绍一种针对积云这种参与介质的可微渲染算法。该算法针对积云的辐射传输 方程进行优化,在保证渲染效果的前提下,提升了渲染效率。

5.4.2 针对积云的可微渲染方法

这里所介绍的可微渲染算法大致分为三个模块。第一个模块是积云的正向渲染过程, 为了提高渲染算法的时间效率,使用光的多次前向散射近似代替光线传播中的多次散射,根 据初始化的密度场、光照、摄像机位置、消光系数等参数,通过体渲染得到二维图像。第二个 模块是渲染参数的梯度计算,根据设定的渲染参数,保存中间过程中每一个中间变量对该场 景参数的导数,并通过链式法则得到二维图像像素值对该参数的导数。最后一个模块是逆 向过程的优化,根据导数以及渲染图像和目标图像的损失值,利用随机梯度下降算法实现场 景参数的优化过程。根据该算法,可以实现了针对光照颜色、吸收系数、散射系数、密度场的 正向渲染及逆向优化过程。

1. 算法实现

可微渲染的总体框架如图 5.14 所示。

可微渲染的主要目的就是利用神经网络来推断场景的渲染参数,从而减少传统算法在 计算渲染参数(这里就是辐射强度)时的开销。整个算法主要包含两个模块,正向渲染与逆 向渲染。首先定义目标图像 I_{target},这张图像是真实的二维图像,目标是找到这张图像对应 的三维场景中的渲染参数 B_{target}。对正向渲染来说,初始化渲染参数为 B_{init},并根据给定的 渲染参数得到一张渲染图像 I_{out}。接着,计算渲染图像与目标图像的误差,并用损失函数表



图 5.14 可微渲染算法的总体流程

示。通过梯度下降算法将损失函数反传到渲染参数上,以确定下一步优化的方向,并根据步 长更新渲染参数 B'。不断重复上述过程,直到损失值达到令人满意的结果。此时的渲染参 数 B'就是可微渲染算法最后优化的结果。

对积云这种参与介质,主要应关注光线的多次前向散射过程。下面分别介绍积云的正 向渲染、逆向渲染,以及优化过程。

1) 积云的正向渲染

多次前向散射过程如图 5.15 所示。给定初始渲染参数,首先通过预计算得到逐网格的 散射强度计算量,然后利用光线行进法进行逐像素的散射强度计算,最终得到渲染图像。





(1) 单次散射的预计算。首先计算单次散射。单次散射模拟光源经过介质在单一方向上的散射,如下列公式所示:

$$J_{\rm ss}(x,\omega) = \frac{\Omega}{4\pi} \int_{4\pi} I_{\rm ri}(x,\omega') p(\omega,\omega') d\omega'$$
(5.12)

 $J_{ss}(x,\omega)$ 是单次散射强度,表示外界光源的光子经过第一次散射就进入到该 x 点的 ω 方向的数量。 $p(x,\omega)$ 指的是该点的相位方程, $I_{ri}(x,\omega)$ 表示该点光线辐射的剩余量。因为每一个点都会受到来自四面八方的单次散射光,所以需要在该点上的所有入射光线进行球面积分,由于球面的立体角积分为 4π ,所以要在最外层除以 4π 。 Ω 表示该点透光率,表示该点通过参与介质的相互作用后,继续沿着 ω 方向行进的辐射量比重。它的数值计算方法由 $\frac{\sigma_s}{\sigma_t}$ 给出,表示了单位距离中光可以被透射出的强度的比例。

将上述积分形式转换为可计算的离散形式,通过构造规则网格,依次计算光线经过的每 个网格的单次散射量,如图 5.16 所示,遍历每个三维网格,求解该点的辐射剩余量。将它的 位置作为起点,它与光源的连线作为方向。根据这个方向与体数据进行求交计算,得到交点 和路径长度。在这段长度之内,依次行进一个个网格,并累加光学厚度。最后,通过光学厚 度、消光系数和光源强度,求解每个像素点的辐射剩余量。

因为在预计算阶段已经计算好了 $I_{ri}(x,\omega)$ 的数据场并保存下来,所以在计算每个点的



图 5.16 辐射剩余量的计算示意图

单散光时,就变得更加高效。渲染过程中,从视点出发,发射多条光线,每条线沿着该方向进 行光线行进,最终的累积量就是每一个像素的辐射强度。

(2) 多次散射项的预计算。为优化计算效率,计算辐射传输方程应借助扩散理论,并通 过近似计算多次散射过程来提高计算效率。按 Nishita 的理论,在云层内部的多次散射过 程中,只有沿着光入射的方向附近的散射光(前向散射光)对于多次散射强度的贡献量最大, 其他方向的贡献可忽略不计。因此,可以简化云的光照模型,近似计算沿着光源方向的散 射,从而减少计算量。

在此前提下,利用辐射传输的扩散理论,可将光线的多次前向散射强度表示为泰勒展开 式的前两项,如下列公式所示:

$$I_{d}(x,\omega) = I_{d}^{0}(x_{u}) + I_{d}^{1}(x_{u}) \cdot \omega$$
(5.13)

其中,I⁰_d(x_u)是各个方向辐射强度的平均值,由一个微分方程表示,如下列公式所示:

$$\nabla k(x_{u}) \cdot \nabla I_{d}^{0}(x_{u}) + k(x_{u}) \cdot \nabla^{2} I_{d}^{0}(x_{u}) - \alpha(x_{u}) I_{d}^{0}(x_{u}) + S(x_{u}) = 0 \quad (5.14)$$

$$k(x_u) = (\sigma_{\rm tr}\rho(x_u))^{-1}$$
(5.15)

$$\alpha(x_u) = \sigma_a \rho(x_u) \tag{5.16}$$

$$S(x_u) = \sigma_s \rho(x_u) Q_{\rm ri}^0(x_u) - \frac{\sigma_s}{\sigma_{\rm tr}} \nabla Q_{\rm ri}^1(x_u)$$
(5.17)

其中, $Q_{ri}^{0}(x_{u})$ 和 $Q_{ri}^{1}(x_{u})$ 是光源辐射剩余量的泰勒展开式的前两项,用于计算多次前向散射,其表示形式如下列公式所示:

$$Q_{\rm ri}^0(x_u) = I_{\rm ri}^0(x_u)$$
(5.18)

$$Q_{\rm ri}^0(x_u) = I_{\rm ri}^0(x_u)$$
(5.19)

 $I_d^1(x_u)$ 表示多次前向散射的主方向分量,如果求解得到了 $I_d^0(x_u)$ 的数据场,就可以进 一步得到 $I_d^1(x_u)$ 的计算结果,其表示形式如下列公式所示:

$$I_{d}^{1}(x_{u}) = k(x_{u})(-\nabla I_{d}^{0}(x_{u}) + \sigma_{s}\rho(x_{u})Q_{ri}^{1}(x_{u}))$$
(5.20)

当得到了 $I_d^0(x_u)$ 和 $I_d^1(x_u)$ 的计算结果,就可以将辐射传输中多次前向散射的积分形 式转换为由这两个变量表示的形式,不再需要进行蒙特卡洛采样并求积分,而是迭代求解离 散形式的微分方程,如下列公式所示:

$$J_{\rm ms}(x_u,\omega) = \frac{\Omega}{4\pi} \int_{4\pi} (I_d^0(x_u) + I_d^1(x_u) \cdot \omega) p(\omega,\omega') d\omega'$$
$$= \Omega I_d^0(x_u) + \frac{\Omega \bar{\mu}}{3} I_d^1(x_u) \cdot \omega$$
(5.21)

由于多次前向散射只需计算沿光线前进方向的散射量,省略了采样和递归的过程,可以 将散射强度的计算放在离线阶段,通过将计算的 $I_d^0(x_u)$ 和 $I_d^1(x_u)$ 存储在每一个网格中。 在实时渲染时,采用光线行进法,当与某个网格相交时,直接调用预计算的数据,使用公 式(5.21)便可求出散射强度。

(3)光线行进法。光线行进法(Ray-Marching)是路径追踪算法中,针对体渲染的特殊 处理方法。这一算法的核心是从视点出发连接成像平面各像素点,向视锥内发射射线,并使 用采样的方法按照一定的规则累积该射线上的辐射强度值,最终渲染在对应像素点上。

对于积云渲染任务,首先定义了一个固定体积的积云密度场,然后利用从视点出的光线 与包围盒求交点,前交点和后交点分别相交于体积的前后两侧,路径为两个交点之间的距 离,按照固定步长一步一步往前计算亮度,并将亮度进行累加,如图 5.17 所示。





在预处理阶段,已经计算好单次散射光和多次前向散射光的数据场,此时可以从视点出发,依次生成射线,将射线与包围盒直接进行求交计算,并在两个交点之间进行射线的 Ray-Marching 过程,将辐射强度进行累加并计算最终该像素的亮度。

2) 积云的逆向渲染

逆向渲染的目标是通过渲染得到的图像推断原始三维场景的几何、光照、纹理、材质、运 动等信息。对积云这种参与介质的渲染,主要需要推断的参数包括光源强度、光源颜色、介 质密度、吸收系数、散射系数等。这些也是正向过程渲染中需要的初始参数。

可导性、连续性是可微渲染的必要条件,算法的关键就在于如何解决体渲染的不可导问题。积云逆向渲染的核心就是求解单次散射与多次散射的导数。

(1) 单次散射项的导数求解。公式(5.12)给出了单次散射的计算方法。通过求导法则 对其进行求导计算,如下列公式所示:

$$\dot{I}_{ss}(x,\omega) = \int_{D_2}^{D_1} (\dot{\tau}(x_u, x_a)k_t(x_u) + \tau(x_u, x_a)\dot{k}_t(x_u))J_{ss}(x_u, \omega)du + \int_{D_2}^{D_1} (\tau(x_u, x_a)k_t(x_u))\dot{J}_{ss}(x_u, \omega)du$$

$$+\dot{D}_{1}\tau(x_{b},x_{a})k_{t}(x_{b})J_{ss}(x_{b},\omega)$$
$$+\dot{D}_{2}\tau(x_{b},x_{a})k_{t}(x_{a})J_{ss}(x_{a},\omega) \qquad (5.22)$$

为了计算 $I_{ss}(x,\omega)$ 的具体导数,首先要求出每个网格单位的单次散射辐射,即 $J_{ss}(x_{u},\omega)$,同理,对其进行求导,并展开如下列公式所示:

$$\dot{J}_{ss}(x_{u},\omega) = \frac{\dot{\Omega}}{4\pi} \int_{4\pi} I_{ri}(x_{u},\omega') p(\omega,\omega') d\omega' + \frac{\Omega}{4\pi} \int_{4\pi} \dot{I}_{ri}(x_{u},\omega') p(\omega',\omega) + I_{ri}(x_{u},\omega') \dot{p}(\omega,\omega') d\omega' + \int_{4\pi} \langle n_{\perp},\dot{\omega}' \rangle p(\omega,\omega') \Delta I_{ri}(x_{u},\omega') dl(\omega')$$
(5.23)

对上面公式中的 $\dot{\tau}(x_u, x_a)$ 进行求导然后展开得到:

$$\dot{\tau}(x_b, x_a) = -\tau(x_b, x_a) \int_{x_a}^{x_b} (\dot{\sigma}_t \cdot \rho(x_u) + \sigma_t \cdot \dot{\rho}(x_u)) du$$
$$-\tau(x_b, x_a) (\dot{D}_1 \cdot \sigma_t \cdot \rho(x_b) - \dot{D}_2 \cdot \sigma_t \cdot \rho(x_a))$$
(5.24)

其中, x_a 和 x_b 分别是从视点方向发射的射线与数据场边界相交求得的前交点(x_a)与后交点(x_b)。根据莱布尼兹-雷诺传输定理(Reynolds transport theorem),如果设n是两个交点的平面的法向量,p是在体数据边界上在交点周围的附近一点, ω 表示光线入射的方向,则D的导数可以由下列公式表示:

$$\dot{D} = \partial_{\pi} \frac{\langle x - p, n \rangle}{\langle \omega, n \rangle}$$
(5.25)

其中, <>表示两个向量之间的内积。莱布尼兹-雷诺传输定理是在积分公式内部进行求导数的莱布尼兹积分的三维表示形式, 主要应用场景为有边界的三维区域的积分项求导数, 其原理就是利用了捕获边界上在求导的位置的周围函数的微小变化求解了该点的微分结果。

最后,消光系数由消光截面和该点的密度的乘积决定,消光截面等于吸收截面和散射截 面的相加。将消光系数的原公式进行展开求导,可以得到下列公式:

$$\dot{k}_{\iota}(x_{u}) = \dot{\sigma}_{\iota}\rho(x_{u}) + \sigma_{\iota}\dot{\rho}(x_{u})$$
(5.26)

最终,解得了单次散射的导数表达式。

(2) 多次散射项的导数求解。针对多次前向散射在路径中的积分,同样对其进行展开。 同样,因为公式中出现了积分公式,使用莱布尼兹-雷诺传输定理,求导展开,如下列公式 所示:

$$\dot{I}_{\rm ms}(x,\omega) = \int_{D_2}^{D_1} (\dot{\tau}(x_u, x_a)k_t(x_u) + \tau(x_u, x_a)\dot{k}_t(x_u))J_{\rm ms}(x_u, \omega)du + \int_{D_2}^{D_1} (\tau(x_u, x_a)k_t(x_u))\dot{J}_{\rm ms}(x_u, \omega)du + \dot{D}_1\tau(x_b, x_a)k_t(x_b)J_{\rm ss}(x_b, \omega) - \dot{D}_2k_t(x_a)J_{\rm ss}(x_a, \omega)$$
(5.27)

其中,对于 $\dot{\tau}(x_u, x_a)$ 的求解参见公式(5.24)。另外,对于 $k_t(x_u)$ 的导数,使用公式(5.26) 对其进行计算,对于边界的导数 \dot{D}_1 和 \dot{D}_2 ,已经在公式(5.25)中进行了解释。最后,只剩下 每个独立的网格单元的多次前向散射的导数,需要进一步计算。 如图 5.18 所示,网格的每一个点由单次散射 J_{ss} 和多次前向散 J_{ms} 组成。多次前向散 射表示为 $I_d^0(x_u)$ 和 $I_d^1(x_u)$ 的线性相加,该算法的核心就是对多次前向散射求导的方法。



图 5.18 多次前向散射的示意图

首先,直接套用求导法则对其进行展开,如下列公式所示:

$$\dot{J}_{\rm ms}(x_u,\omega) = \dot{\Omega}(I_d^0(x_u) + \frac{\bar{\mu}}{3}I_d^1(x_u) \cdot \omega) + \Omega(\dot{I}_d^0(x_u) + \frac{\bar{\mu}}{3}\dot{I}_d^1(x_u) \cdot \omega + \frac{\bar{\mu}}{3}I_d^1(x_u) \cdot \omega)$$
(5.28)

其中, Ω 的求解方法较为简单, 直接使用求导法则对其进行展开, 如下列公式所示:

$$\dot{\Omega} = \frac{\sigma_t \dot{\sigma}_s - \sigma_s \dot{\sigma}_t}{(\sigma_t)^2} \tag{5.29}$$

求解多次前向散射的导数是本算法的关键,核心在于求解 I_d^0 和 I_d^1 的导数,同样直接 在原式中求导数,可以得到下列公式:

$$\nabla \dot{k}(x_{u}) \cdot \nabla I_{d}^{0}(x_{u}) + \nabla k(x_{u}) \cdot \nabla \dot{I}_{d}^{0}(x_{u}) + \dot{k}(x_{u}) \cdot \nabla^{2} I_{d}^{0}(x_{u}) + k(x_{u}) \cdot \nabla^{2} \dot{I}_{d}^{0}(x_{u}) - \dot{a}(x_{u}) \dot{I}_{d}^{0}(x_{u}) + S(x_{u}) = 0$$
(5.30)

对于 I_d^0 的梯度,可以使用数值的方法对其进行求解,因为数据场以三维网格的形式给出,在已知 I_d 的数据场的基础上,就可以使用结构动力学中的中心差分法(利用有限差分求解一阶导数)得到辐射强度 I 的梯度,即当前网格与周围网格的差值就是当前网格的梯度,如下列公式所示。

$$\nabla I = \frac{I_{i+1,j,k} + I_{i-1,j,k} + I_{i,j+1,k} + I_{i+1,j,k} + \dots + I_{i,j,k-1} - 6I_{i,j,k}}{h^2} \qquad (5.31)$$

在公式(5.31)中,*h* 表示网格的单位长度,*i*、*j*、*k* 分别表示某网格的长、宽、高的坐标。 只要对该网格的亮度与周围 8 个网格的亮度做差,即可得到梯度的数值解。

最后,由于 $I_d^1(x_u)$ 是根据 $I_d^0(x_u)$ 求解得来,如果已知 $I_d^1(x_u)$,那么就可以直接使用 求导法则求出 $I_d^1(x_u)$ 的导数,如下列公式所示:

$$\dot{I}_{d}^{1}(x_{u}) = \dot{k}(x_{u})(-\nabla I_{d}^{0}(x_{u}) + \sigma_{s}\rho(x_{u})Q_{ri}^{1}(x_{u})) + k(x_{u})(-\nabla \dot{I}_{d}^{0}(x_{u}) + \dot{\sigma}_{s}\rho(x_{u})Q_{ri}^{1}(x_{u}) + \sigma_{s}\dot{\rho}(x_{u})Q_{ri}^{1}(x_{u}) + \sigma_{s}\rho(x_{u})\dot{Q}_{ri}^{1}(x_{u}))$$
(5.32)

综上所述,此时分别得到了光的单次散射和多次前向散射对场景参数的导数,通过计算

这些中间变量对场景参数的导数,建立了渲染图像与场景参数的联系。

(3) 基于梯度下降的优化。下式表示了渲染参数的优化过程:

$$B_{\text{new}} = B - \gamma \cdot \frac{\partial \left(\| I_{\text{out}}(B) - I_{\text{target}} \| \right)}{\partial B}$$
(5.33)

其中, *I*_{out}(*B*)表示根据当前渲染参数 *B* 渲染出来的图像, *I*_{target} 表示目标图像, γ 表示梯度 下降的步长。其中,使用渲染图像和目标图像的均方差作为损失函数。

2. 实验结果

1) 实验参数设置

本算法渲染的三维体数据规模为 64 像素×64 像素×96 像素,渲染出的二维图像的分 辨率为 512 像素×512 像素,每个实验优化迭代次数为 100 次。本文使用的烟雾数据是基 于物理模拟生成的,其余模型来自斯坦福扫描模型,并使用 Binvox 工具转换为体素,并生成 三维密度场数据。

2) 正向渲染的实验结果

图 5.19 展示了不同积云密度场的渲染效果。



图 5.19 积云正向渲染结果

可以看出,所设计的正向渲染模块能够较好地保留细节信息。正向渲染的时间开销如图 5.20 所示。



可以看出,除第一次求解需要 5.3s 外,平均求解速度为 1.83s。相比基于蒙特卡洛估计的体渲染,渲染效率得到显著提升。

3) 逆向渲染的实验结果

可微渲染除了可以高效地进行正向渲染外,还可通过指定场景参数的方式进行特定参



数的优化。图 5.21 展示了逆向渲染对场景参数的优化效果。

如图 5.21 所示,给定目标图像后,所设计的逆向渲染经过 100 次的迭代,已经可以很好 地逼近真实图像。此外,逆向渲染对于场景参数优化的效率也较高,平均 100 次迭代之后基 本可以得到最终优化结果,总优化时间在 100s 以内。

5.5 习题

1. 增强现实与虚拟现实有何不同? 二者在关键技术方面的差异主要在哪里?

2. 智能化对虚拟现实与增强现实各有哪些影响?

3. 人工智能生成三维内容可以优化建模的哪些方面?对渲染和交互能起到什么作用?

图 5.21 逆向渲染优化场景参数效果图