

# 浙 江 大 学

## 硕士学位论文文献综述

(专业学位)

文献综述题目： 三维服装线上展示文献综述

姓 名： 李娜

学 号： 21951471

专 业： 软件工程

院 别： 软件学院

导 师： 金小刚

日期：二零二零年 十一月

# 目 录

|                      |       |
|----------------------|-------|
| 一、 Web3D 模型问题概述..... | - 1 - |
| 二、 WebGL 研究现状.....   | - 1 - |
| 三、 三维服装线上展示相关技术..... | - 2 - |
| 3.1 三维模型减面压缩.....    | - 2 - |
| 3.2 PBR 渲染.....      | - 4 - |
| 3.3 透明渲染.....        | - 5 - |
| 3.4 抗锯齿处理.....       | - 7 - |

## 一、 Web3D 模型问题概述

通过分析,将 Web3D 模型渲染的问题分为两个方面:模型的真实性和模型加载与渲染的速度。模型的真实性的是指在 Web 端渲染 3D 模型时,模型渲染后的效果能够与真实的自然界中所存在的车所表现出的外观效果一致或极为接近。目前影响渲染效果的主要因素为 Web 端较弱的渲染能力以及在 Web 端渲染模型时所涉及到的大量的计算过程<sup>[1]</sup>。

## 二、 WebGL 研究现状

Web3D 技术是随着互联网和虚拟现实技术发展而产生的<sup>[2]</sup>。随着浏览器的版本升级和能力提升,传统的插件式三维如 osgEarth、skyline 等已经渐渐退出历史舞台,以 OpenGL 渲染为主的机制也逐渐过渡到 JavaScript 和 OpenGL ES 2.0 相融合的 WebGL 渲染。依托其优越的着色渲染机制,WebGL 成为客户端 Web3D 三维场景渲染的首选<sup>[3]</sup>。

WebGL 的优势主要体现在以下 4 点:

### (1)无插件

相比传统的 osgEarth 三维地球,系统仅支持通过安装插件运行在 IE 内核的浏览器,用户体验差,门槛高;WebGL 实现了多个主流浏览器(如 Chrome、Firefox)的无插件浏览。

### (2)兼容性好

当前主流的浏览器 Chrome、Firefox、Safari 对 WebGL 都有很好的兼容性,IE11 浏览器除了不支持场景的坐标拾取,对 WebGL 的三维场景浏览也表现出了较好的

兼容性。

### (3)跨平台

OpenGL 是统一、标准、跨平台的应用程序编程接口,以 OpenGL ES 2.0 为基础的 WebGL 同样实现了跨平台支持。

### (4)硬件加速

相比传统三维引擎,WebGL 在基于客户端渲染的同时,实现了系统的硬件加速,使网页渲染直接调用显卡 GPU 加速,保障在三维场景渲染时充分利用客户端资源,提高三维渲染效率,提升用户体验,也是驱动 WebGL 三维发展的主要原因之一。

## 三、 三维服装线上展示相关技术

### 3.1 三维模型减面压缩

三维模型减面压缩在很多情境中都是非常重要的。模型生成时可能被用于各种各样的目的,为了达到最好的效果,模型构建者往往倾向于使用更多的面片将模型制作得更加精致逼真。而在使用时,为了适应系统的处理能力,在有限的资源条件下提高模型的渲染速度,提升用户的视觉效果体验,人们提出了很多三维模型减面压缩算法<sup>[4]</sup>。

常见的模型减面算法主要包含以下几种类型:顶点聚类(Vertex Clustering),增量式简化(Incremental Decimation),采样(Sampling)和自适应细分(Adaptive Subdivision)。

顶点聚类是由 Rossignac 与 Borrel 最早提出<sup>[5]</sup>的,顾名思义就是使用聚类算法对顶点进行重新整理。在实现时,首先需要将三维模型所在的空间分为一个个小格子(小格子的尺寸会影响到聚类的效果和速度,因此需要寻找一个合适的值

来保持简化质量和速度之间的均衡），之后通过聚类算法为每一个小格子都计算出一个代表顶点，这个顶点作为简化模型中该格子中的唯一顶点，与其他相邻格子相连构建出新的边和面片。在进行模型转化时，原始模型中同一格子的顶点均会被合并，成为新顶点，原始顶点之间的相连边和面片也会被删除，因此模型最终会得到简化。顶点聚类是一个  $O(n)$  的算法<sup>[6]</sup>。

顶点聚类算法不会受到原始模型拓扑结构的限制。无论模型顶点面片是怎样的连接关系，该算法都可以进行简化操作。然而，在使用顶点聚类进行模型简化时，由于之前的顶点、边和面片均可能被替换或者删除，因此网格的拓扑结构会随之改变。拓扑结构的改变虽然可以对网格进行大幅度的简化，然而在简化过程中，可能会使模型的微小特征消失，使得外观产生变化。该变化的大小则取决于小格子的尺寸。小格子尺寸越小，简化得到的模型会保留越多细节特征，然而相对的，简化的幅度可能就会越小。在已定小格子大小的情况下，顶点聚类也很难达到最优。因为模型的拓扑结构并不一定是均匀的，计算较大的平面时一般可以删除更多的顶点，而尖锐的特征则很可能无法保留<sup>[22]</sup>。

增量式删除中“增量”指的是对模型进行一系列微小的简化操作，在每一步操作中都保证会降低模型的复杂度（如点删除、边删除等等），同时也会减少模型的精细程度<sup>[7]</sup>。其核心思想是每步操作都删除一个被选中的顶点，删除的同时需要注意更新该顶点所在的边及面片。通常采取的操作是将该顶点所在的所有面片删除，并构建新的三角面片对删除造成的空洞进行填补。边折叠由 Hoppe 等人<sup>[8]</sup>首次提出，指的是每次删除一条边，包括这条边上的两个顶点以及该边相邻的两个三角面片，同时选取合适的位置将两个顶点合并为一个新顶点。

整体来看，增量式删除算法的计算复杂度是比顶点聚类算法高一些的。由于

需要排序，因此平均需要  $O(n\log n)$ ，最坏时可能达到  $O(n^2)$ 。然而，由于对简化质量进行了评估，因此简化后得到的模型质量相对会更高一些。同时，由于增量式删除是由一系列很小的简化步骤组合而成，因此在过程中可以很方便地获得不同精细程度的模型，用于不同场景中的需求。本系统最终也实现并使用了一种基于边折叠的增量式删除减面算法<sup>[9]</sup>。

采样算法的思想十分简单。首先需要在原始模型中定义一种方式计算需要的采样点，将采样点提取出来之后构建新的三角面片，进而生成简化后的模型。采样点的密度和数量会影响简化后模型的质量，越密集的采样点保留的原始信息就越多，然而简化程度也就越低。现有的算法中有使用排斥力计算采样<sup>[10]</sup>，基于体素（Voxel）提取采样点<sup>[11]</sup>等等。

### 3.2 PBR 渲染

PBR（Physically Based Rendering），翻译为中文即是基于物理属性的渲染<sup>[12]</sup>。近年来越来越多的游戏或电影公司开始使用 PBR 渲染技术进行游戏或电影的制作。由于 PBR 是基于物体的物理本质原理来对场景进行真实感渲染，因此在虚拟场景中可以更加逼真地模拟出现实中的效果。随着虚拟现实和增强现实产业的发展，大量的应用也开始使用 PBR 渲染技术<sup>[13]</sup>。图 1 中列举了基于 PBR 的电影截图以及 HTML5 中的应用截图。



图 1 HTML 中的 PBR 算法应用

PBR 算法是一个基于实物的阴影系统但却不同于在其之前的阴影系统、遵守能量守恒定律，并证明了扩散、反射与能量守恒就是 PBR 算法的核心思想<sup>[14]</sup>。PBR 渲染算法与传统的渲染方式相同，通过着色模型和光照模型两者相结合来模拟出比较真实的视觉效果。在传统的渲染算法中，主要使用的是一些自定义数量、类型的理想光源，同时结合需要的贴图来实现渲染效果。这些算法需要基于设计人员自身的尝试和经验来向想要的场景靠近。在他们进行设计时，可能并不十分理解每个参数的真正意义，而是靠着不同参数的调整和经验积累来努力将场景制作得更加逼真。比如根据生活经验，我们可以知道金属材质对光线的反射能力比较强，那么在模拟金属材质时，就使用值比较大的高光贴图。相反，对于毛线、磨砂等材质就会使用比较弱的高光贴图。因此，传统的渲染方法对美工设计人员的专业素养要求比较高<sup>[15]</sup>，需要具有丰富的生活经验并熟悉不同的材质属性。

### 3.3 透明渲染

在现实生活中，具有半透明属性的物体是十分常见的，如我们人类的皮肤，透过皮肤我们可以隐约可见下方存在的静脉血管。在服装渲染的过程中，薄纱、防晒服等等都具有半透明属性，为了更真实地对服装进行模拟渲染，本系统参考了现有的半透明渲染算法进行实现并提出了优化。在渲染领域中，如何实时快速地进行半透明渲染一直是一个不断探索的热门问题<sup>[16]</sup>。Web 端常见的透明渲染算法主要包括以下几种：透明度混合（Alpha Blending），基于权重的混合算法（Weighted Blended OIT），基于深度剥离的透明算法（Depth Peeling），基于矩的透明算（Moment Based OIT）。

透明度混合是现在 OpenGL 中最简单最常见的方法。在设置颜色值时，除了需要使用 RGB 值表示当前片元的颜色，还可以通过 Alpha 通道设置片元的透明



度。默认情况下该通道的值为 1，即片元不透明，若完全透明则值为 0。在渲染计算时，屏幕像素点的颜色会存储在颜色缓存（Color Buffer）中。而对于半透明的物体，则需要根据透明度来将颜色值与当前的颜色缓存进行混合，使得用户肉眼看上去好像透明一般。因此该算法称为透明度混合。

基于权重的混合算法是由 McGuire 等人<sup>[17]</sup>提出的，全称为基于权重混合的顺序无关透明算法。OIT 即是 Order Independent Transparent，顺序无关的透明。与透明度混合算法不同，基于权重的混合算法不需要对半透明物体或者图元（primitive）或者片元（fragment）进行排序。这种算法希望通过避免排序，来缩减开销，提升渲染速度。

在 2001 年，Nvidia 提出了深度剥离算法，在 Everitt 的报告<sup>[18]</sup>中有着详细的阐述，其需要通过深度比较将已经计算的好内容一层一层剥离，以获得更深层的物体内容。从相机向某一个方向射出的射线，会与很多物体相交。如果最近的相交点是半透明的，那么我们就需要获得第二近的相交点，并将其与原先得到的颜色值进行混合。以此类推，若某一层的相交点是半透明状态，我们就需要继续计算下一层的相交点并混合。每获得一层的颜色值，就需要进行一遍渲染。

基于矩的透明算法与基于权重的混合算法相同，都是没有使用深度排序的顺序无关透明算法。矩（Moment）在算法中指的是一个概率密度函数  $p(x)$  通过另一个函数  $b(x)$  进行积分所得到的值。而基于矩的透明算法的核心就是通过矩来还原出概率分布函数  $P(x)$  的累计分布函数<sup>[19]</sup>。这种算法希望借助矩的计算替代权重混合算法中的权重函数，以获得更好的混合方程。在实现基于矩的顺序无关透明算法时，首先需要将不透明的物体渲染出来。然后开始绘制半透明物体，并使用矩的计算公式来获得矩。之后根据得到的结果计算出吸收度与穿透度，最后



进行颜色混合。基于矩的透明算法是由 Muenstermann 等人<sup>[20]</sup> 于 2018 年提出的方法，目前暂未得到广泛的应用。

### 3.4 抗锯齿处理

在渲染的场景中，很多游戏和电影都已经可以以假乱真，然而在追求光照与材质真实感的同时，我们也要注意渲染的细节。在部分游戏中，放大场景后可以发现部分物体边缘并不十分平滑，而是以一种锯齿的方式呈现。这是因为渲染时我们需要将三维的场景转化为二维屏幕上的像素点，而在转换的过程中，可能有一些像素点覆盖在了不同网格的边缘上，因此需要做一个取舍，这个取舍的选择就导致了走样（Aliasing）现象。而平滑边缘，降低走样程度的处理过程就被称为抗锯齿处理（Anti-aliasing）。

走样现象主要是由分辨率的限制引起的，而抗锯齿处理的核心思想就是对锯齿部分进行柔化处理，来使色块边缘变得平滑，削弱图像的锯齿感。当前比较常见的几种抗锯齿算法包括超级采样抗锯齿（SSAA），多重采样抗锯齿（MSAA），形态抗锯齿（MLAA）和快速近似抗锯齿（FXAA）等。

超级采样抗锯齿（Super-Sampling Anti-Aliasing）<sup>[23]</sup>是抗锯齿方法中比较暴力比较简单易懂的一种方法。在实现时，需要先先在缓存中放大图像，然后对每个像素点进行采样（可以均匀采样，随机采样，抖动采样等），最后对所有采样点的颜色计算平均值作为其显示的结果。这种抗锯齿算法可以使最终的颜色值包含邻近像素的颜色特征，提供给用户过渡的视觉效果。但由于每一帧都需要进行抗锯齿计算，开销会比较大。

在超级采样抗锯齿中，针对每一帧的每一个像素点都需要执行一次采样并保存，因此时间和空间的开销都非常大。为了对 SSAA 进行优化，OpenGL 提出

了多重采样抗锯齿（Multi-Sampling Anti-Aliasing）。在 MSAA 中，只针对深度缓存和模板缓存中的像素点进行处理，也就是说只有图像的边缘——需要进行柔和操作的像素点才会进行采样计算，因此大大缩减了抗锯齿的开销。

形态抗锯齿（Morphological Anti-Aliasing）<sup>[24]</sup>与前两种算法不同，是一种在后处理中进行边缘平滑处理的技术。MLAA 的主要思想是将图像边缘分为多个 L 形状的组合，然后针对 L 形状进行抗锯齿处理。首先比较每个像素及其相邻像素找到像素间的中断，中断一般分为横向和纵向，并对中断像素添加标记。然后遍历像素点，通过有中断标记的像素识别中断线，并进行组合。由于中断线只有横向和纵向，因此组合后的形状可被分成 L 形、U 形和 Z 形，其中 U 形和 Z 形也可以用两个 L 形拼合表示。最后使用针对 L 形的混合算法进行抗锯齿处理。因为该算法是基于图像边缘的形态进行计算，所以称为形态抗锯齿算法。

快速近似抗锯齿（Fast Approximate Anti-Aliasing）算法<sup>[25]</sup>与形态抗锯齿比较相似，也是在后处理中执行的抗锯齿算法，它的特点是不需要硬件的支持，因此对显卡没有要求。FXAA 在实现时首先也需要进行边缘检测，通过比较像素的亮度信息来查找需要处理的像素。与 MLAA 不同的是 FXAA 只识别较长的边，而不会识别边缘的形状。在得到长边后，通过像素点与边求交来近似于像素的覆盖率，最后根据覆盖率将颜色值混合。与其他抗锯齿算法相比，由于忽略了非边缘像素的反走样，抗锯齿 FXAA 的效果会较差一些，但速度较快且可用于图像处理。为了保证渲染的实时性，本系统选择了 FXAA 进行实现。

## 参考文献

- [1] 郑维欣,贾金原.基于 PBR 的轻量级 WebGL 实时真实感渲染算法[J].系统仿真学报,2017,29(11):2693-2699+2708.
- [2] 王维敏. Web3D 技术探索及几种 Web3D 技术的比较选择[D].武汉大学,2004.
- [3] 赵军,官丽玮,周圣川,胡振彪,王海银.基于 WebGL 的智慧城市三维地理信息系统设计与实现[J].城市勘测,2020(04):44-49.
- [4] 郭力真, 吴恩华. 多边形模型简化算法综述 [J]. 计算机应用研究, 2005, 22(8): 20 – 23.
- [5] Rossignac J, Borrel P. Multi-resolution 3d approximations for rendering complex scenes[C].Modeling in Computer Graphics, 1993: 455 – 465.
- [6] 林俊山,温晓芳,陈梅.一种基于核心顶点的无参图聚类算法[J].计算机应用研究,2018,35(12):3598-3602.
- [7] Cignoni P, Montani C, Scopigno R. A comparison of mesh simplification algorithms[J]. Computers & Graphics, 1998, 22(1): 37 – 54.
- [8]
- [9] Hoppe H, DeRose T, Duchamp T, et al. Mesh optimization[C]. Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, 1993: 19 – 26.
- [10] Garland M, Heckbert P S. Surface simplification using quadric error metrics[C]. Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, 1997: 209 – 216.
- [11] Turk G. Re-tiling polygonal surfaces[J]. ACM SIGGRAPH Computer Graphics, 1992, 26(2):55 – 64.
- [12] Hoppe H. New quadric metric for simplifying meshes with appearance attributes[C]. IEEE Visualization, 1999: 59 – 66.
- [13] Miyashita L, Ishihara K, Watanabe Y, et al. Zoematrope: a system for physical

- material design[J].ACM Transactions on Graphics (TOG), 2016, 35(4): 66.
- [14]郑维欣, 贾金原. 基于 PBR 的轻量级 WebGL 实时真实感渲染算法 [J]. 系统仿真学报, 2017,29(11): 2693 – 2699.
- [15]Boris Raymond, Gaël Guennebaud, Pascal Barla. Multi-scale rendering of scratched materials using a structured SV-BRDF model. 2016, 35(4):1-11.
- [16]Zhao N, Cao Y, Lau R W H. What characterizes personalities of graphic designs?[J]. ACM Transactions on Graphics (TOG), 2018, 37(4): 116.
- [17]吴博剑. 基于图像的透明物体重建与渲染[D].中国科学院大学(中国科学院深圳先进技术研究院),2019.
- [18]McGuire M, Bavoil L. Weighted blended order-independent transparency[J]. Journal of Computer Graphics Techniques, 2013, 2(2): 122 – 141.
- [19]Everitt C. Interactive order-independent transparency[R]. Technical report, NVIDIA Corporation, 2001.
- [20]Shi Rong,Napel Sandy,Rosenberg Jarrett K,Shin Lewis K,Walsh Cynthia F,Mogensen Monique A,Joshi Ajay Jayant,Pankhudi Pankhudi,Beaulieu Christopher F. Transparent rendering of intraluminal contrast for 3D polyp visualization at CT colonography.[J]. Journal of computer assisted tomography,2007,31(5).
- [21]Münstermann C, Krumpfen S, Klein R, Peters C. Moment-based order-independent transparency[C]. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2018:1 – 20.
- [22]林俊山,温晓芳,陈梅.一种基于核心顶点的无参图聚类算法[J].计算机应用研究,2018,35(12):3598-3602.
- [23]Siegl C, Meyer Q, Sußner G, Stamminger M. Solving aliasing from shading with selective shader supersampling[J]. Computers & Graphics, 2013, 37(8): 955 – 962.
- [24]Reshetov A. Morphological antialiasing[C]. Proceedings of the ACM SIGGRAPH / EURO GRAPHICS Conference on High Performance Graphics, 2009: 109 – 116.
- [25]Jimenez J, Gutierrez D, Yang J, et al. Filtering approaches for real-time anti-aliasing[J]. ACM SIGGRAPH Courses, 2011, 2(3): 4.